

 [Informatik 6bi Schuljahr 2019/2020 als PDF exportieren](#)

Informatik 6. Klasse - Schuljahr 2019/20

Lehrinhalte

- [Lehrplaninhalte](#)

[Remote-Zugriff auf Schulserver](#)

Exkursion

- Apple Workshop zu Videoschnitt (September/Oktober)

Projekt

- Projekt zu Videoschnitt (Oktober/November)

Kapitel

- [1\) Audiotbearbeitung](#)
- [2\) Videobearbeitung](#)
- [3\) Web-Techniken \(HTML & CSS\)](#)
- [4\) Netzwerke](#)
- [5\) Textverarbeitung](#)
- [6\) Tabellenkalkulation](#)
- [7\) C++ Arrays, Sortieralgorithmen, Strukturen & Klassen](#)
- [8\) C# Visuelles Programmieren](#)
- [9\) 3D-Modellierung](#)

Leistungsbeurteilung

- **Test (SA)**
 - 2x Tests pro Semester
- **Mitarbeit (MA)**
 - Aktive Mitarbeit im Unterricht (aMA)
 - Mündliche Stundenwiederholungen (mMA)
 - Schriftliche Stundenwiederholungen (sMA)
- **Praktische Arbeiten (PA)**
 - 1x praktischer Arbeitsauftrag pro Woche
- [Aktueller Leistungsstand](#)

Stoff für den 4. Test in Informatik - 6bi - 12. Mai

Stoff für den 3. Test in Informatik - 6bi - 24. März

Stoff für den 2. Test in Informatik - 6bi - 21. Jänner

Kapitel 3 & 4 (inkl. Übertragungsmedien)

Stoff für den 1. Test in Informatik - 6bi - 20. November

Kapitel 1 bis inkl. 3.4.4.7

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920

Last update: **2020/05/28 20:06**



Was wird in der 6. Klasse gemacht?

3. Semester

Sicherung der Nachhaltigkeit

- Notwendiges Vorwissens für die Kompetenzbereiche dieses Moduls wiederholen und aktivieren
- Grundlagen für die Kompetenzbereiche dieses Moduls ergänzen und bereitstellen

Gesellschaftliche Aspekte der Informationstechnologie

Kommunikation und Kooperation

- Cloud-Systeme kennen und einsetzen können
- Vor- und Nachteile von Cloud-Systemen bewerten können
- Online-Informationssysteme sinnvoll nutzen können

Verantwortung, Datenschutz und Datensicherheit

- Maßnahmen und rechtliche Grundlagen im Zusammenhang mit Datensicherheit, Datenschutz und Urheberrecht kennen und anwenden können.
- Social-Media-Zugängen sicher einrichten können
- Verantwortungsvoll mit Social-Media-Angeboten umgehen können

Informatiksysteme - Hardware, Betriebssysteme und Vernetzung

Virtualisierung

- Grundkonzepte der Virtualisierung kennen, Vor- und Nachteile der Virtualisierung angeben können
- Virtualisierungs-Software kennen und einsetzen können

Betriebssysteme und Software (Linux, MacOS, iOS, Android)

- Das Betriebssystem Linux kennen und installieren können
- Mit graphische Oberflächen (KDE / Gnome) von Linux umgehen können
- Grundlegende Konsolenkommandos von Linux beherrschen
- Die Charakteristika von MacOS, iOS, Android kennen

Algorithmik und Programmierung

Makros und Scripts in Anwenderprogrammen

- Office-Software einstellen und adaptieren können
- Scripts und Makros in Office-Software erstellen können

Angewandte Informatik, Datenbanksysteme und Internet

Kalkulationsmodelle und Visualisierung

- Standardsoftware für Kalkulationen und zum Visualisieren anwenden können.
- Kalkulationsmodelle erstellen und die Ergebnisse bewerten und interpretieren können
- Kalkulationsmodelle aus verschiedenen Bereichen (Biologie, Physik, Finanzmathematik, Systemdynamik etc.) erstellen, bewerten und interpretieren können
- Komplexere Tabellenkalkulationsfunktionen kennen und einsetzen können
- Script-Programmierung in der Tabellenkalkulation beherrschen

Datenmodelle und Datenbanksysteme

- Datenbanken im Direktbetrieb verwenden können
- Datenbanken benutzen und einfache Datenmodelle entwerfen können
- Ein bestehendes Datenmodell in einer Datenbank abbilden können
- Unterschiedliche Arten der Beziehung unterscheiden können
- Datenbankabfragen ausführen können
- Berichte, Berechnung und Gruppierungen ausführen können

4. Semester

Sicherung der Nachhaltigkeit

- Notwendiges Vorwissens für die Kompetenzbereiche dieses Moduls wiederholen und aktivieren
- Grundlagen für die Kompetenzbereiche dieses Moduls ergänzen und bereitstellen

Algorithmik und Programmierung

Algorithmen und Datenstrukturen

- Algorithmen erklären, entwerfen, darstellen können.
- Strukturen kennen und einsetzen können
- Ein- und mehrdimensionale Arrays kennen und einsetzen können

- Einfache Sortieralgorithmen kennen
- Wichtige und bekannte Algorithmen aus verschiedenen Gebieten kennen.

Programmierung (Objektorientierte visuelle Programmiersprache)

- Grundlegende Komponenten einer visuellen Programmiersprache kennen und mit ihnen arbeiten können
- Die Grundkonzepte einer visuellen Programmiersprache kennen
- Algorithmen in einer objektorientierten visuellen Programmiersprache implementieren können
- Mit obigen Datenstrukturen und Algorithmen in der visuellen Programmiersprache arbeiten

Angewandte Informatik, Datenbanksysteme und Internet

Textverarbeitungs- und Satzsysteme

- Das Satz-System LaTeX kennen und anwenden können
- Mit einem Formelsatz in einem Text- und Satzsystem arbeiten können

Web-Techniken

- Fortgeschrittene HTML-Techniken (Formulare, CSS, XML) kennen und einsetzen können
- Grundlagen der Script-Programmierung in PHP am Webserver beherrschen
- Kontrollstrukturen (Verzweigungen, Schleifen) kennen und einsetzen können
- Indizierte und assoziative Arrays kennen und anwenden können
- Funktionen kennen und definieren können

=> 6. Klasse (3 Stunden, 2-3 Tests pro Semester)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:0_lehrplaninhalte

Last update: 2019/09/06 20:55



1) Audiotbearbeitung

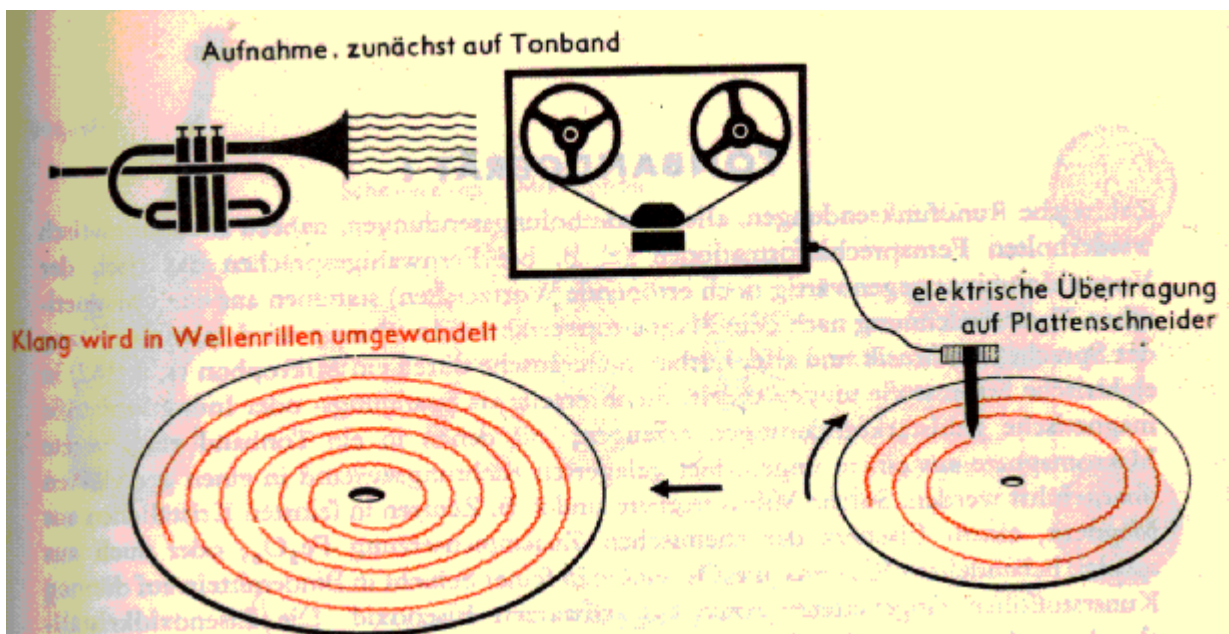
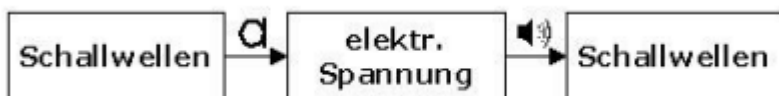
1.1) Theorie

Schallwellen

Schallwellen sind Druckschwankungen in einem Übertragungsmedium (z.B. Luft, Wasser), die von uns wahrgenommen werden können.

In der Analogtechnik wird der Schalldruck durch eine dazu proportionale elektrische Spannung repräsentiert.

- ein Mikrofon wandelt den Schalldruck in die elektrische Spannung um
- diese Spannung kann in andere analoge Darstellungsformen überführt werden
 - Magnetisierung (auf Tonband)
 - Ausleuchtung (auf die Tonspur beim Film)
 - mechanische Auslenkungen (zur Erstellung einer Schallplatte)
- am Ende übersetzt ein Lautsprecher die Spannung wieder in einen dazu analogen Wechseldruck



Ton

Ein Ton ist eine Schwingung, die durch periodische Regelmäßigkeit eine definierbare Grundfrequenz - die Tonhöhe - hörbar werden lässt. Der Klang eines Tons hängt von den Anteilen weiterer Schwingungen innerhalb der Grundschwingung ab. Ein Geräusch - etwa einer Rassel - hat dagegen keine periodische Grundschwingung, die Luftdruckschwankungen verlaufen so unregelmäßig, dass

keine Tonhöhe definierbar ist. Die Übergänge zwischen Ton und Geräusch sind fließend, denkt man daran, dass sich Konzertpauken sehr wohl stimmen lassen, obwohl sie als perkussive Instrumente bereits sehr diffuse Klangspektren erzeugen.

Sound

Sound wird heute oft im 'Multimedia-Deutsch' für alles, was mit dem Audibereich zu tun hat verwendet. Zum Teil geschieht das sicher aus Unkenntnis der Begriffe aus Akustik und Musik. Der Begriff **Sound** hat jedoch auch Vorteile, da er problematische Fragen der traditionellen Begriffe und Wissenschaftsdisziplinen ausblendet (Was ist Musik?!).

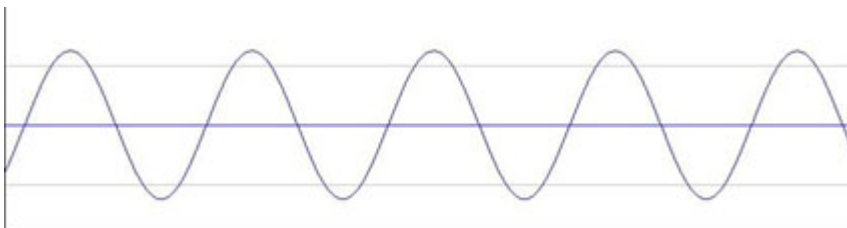
Sprache

Sprache ist danach ein geräuschhaftes=nichtperiodisches Klangereignis, das durch seine definierten Frequenzgemische sprachliche Lautmuster enthält. Im Gesang werden diese Lautmuster einer periodischen Schwingung aufmoduliert. Für gute Sprachverständlichkeit reicht ein Frequenzbereich bis 8 kHz vollkommen aus.

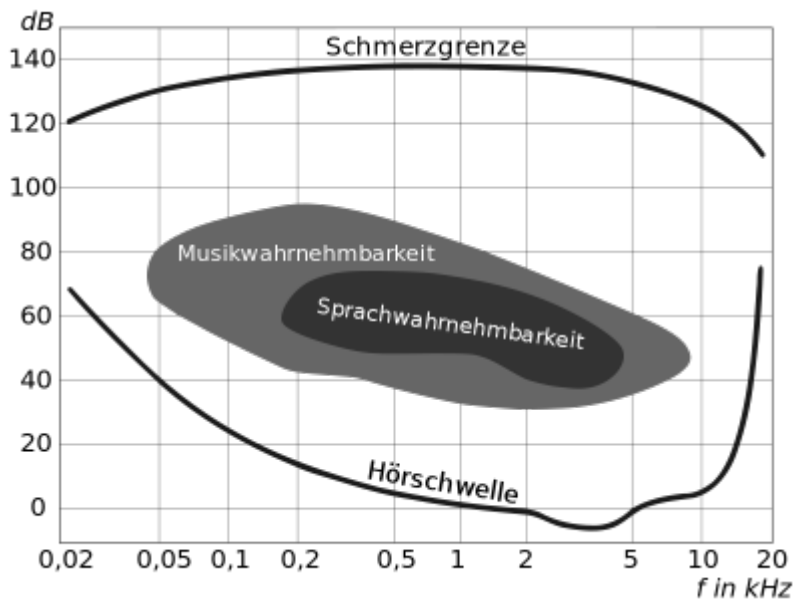
Beispiele für Schwingungen im Audibereich

Sinuswelle

Das Urbild einer periodischen Schwingung, sie besitzt keine weiteren Frequenzanteile.



Hörbereich



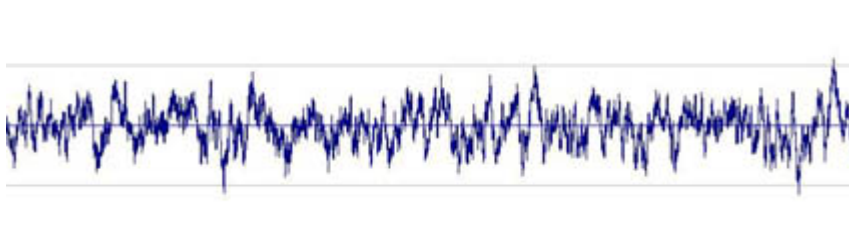
Sinuswellenton in verschiedenen Frequenzen → Teste dein Gehör!



Sinuswelle in verschiedenen Frequenzen (20Hz bis 20kHz)

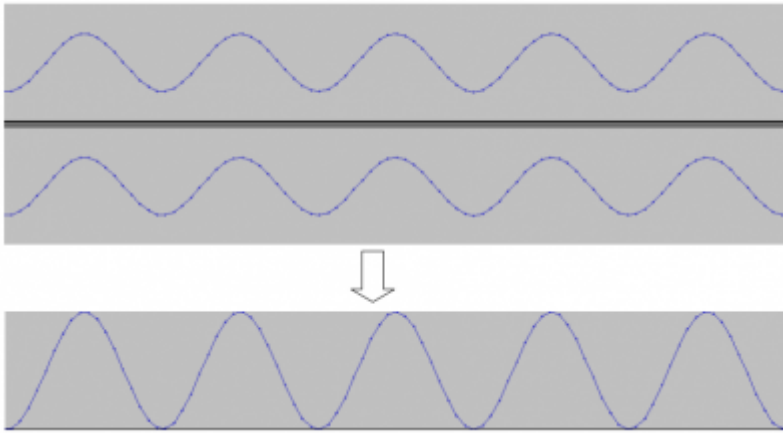
Rauschen

Schwingungschaos = das Gegenteil einer periodischen Schwingung.

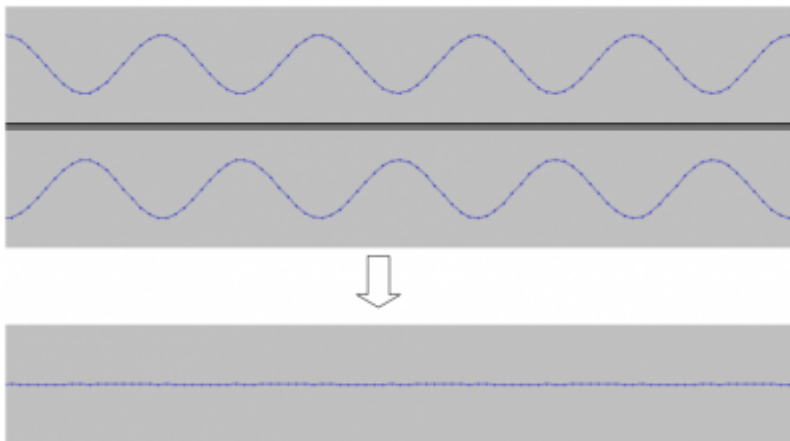


Kombination von Tonquellen

Addition von ähnlichen (kohärenten) Signalen führt zu einer Verstärkung der Amplitude.



Addition von zeitlich verschobenen Tonsignalen führt zu einer Reduzierung der Amplitude.



Maskierungseffekt



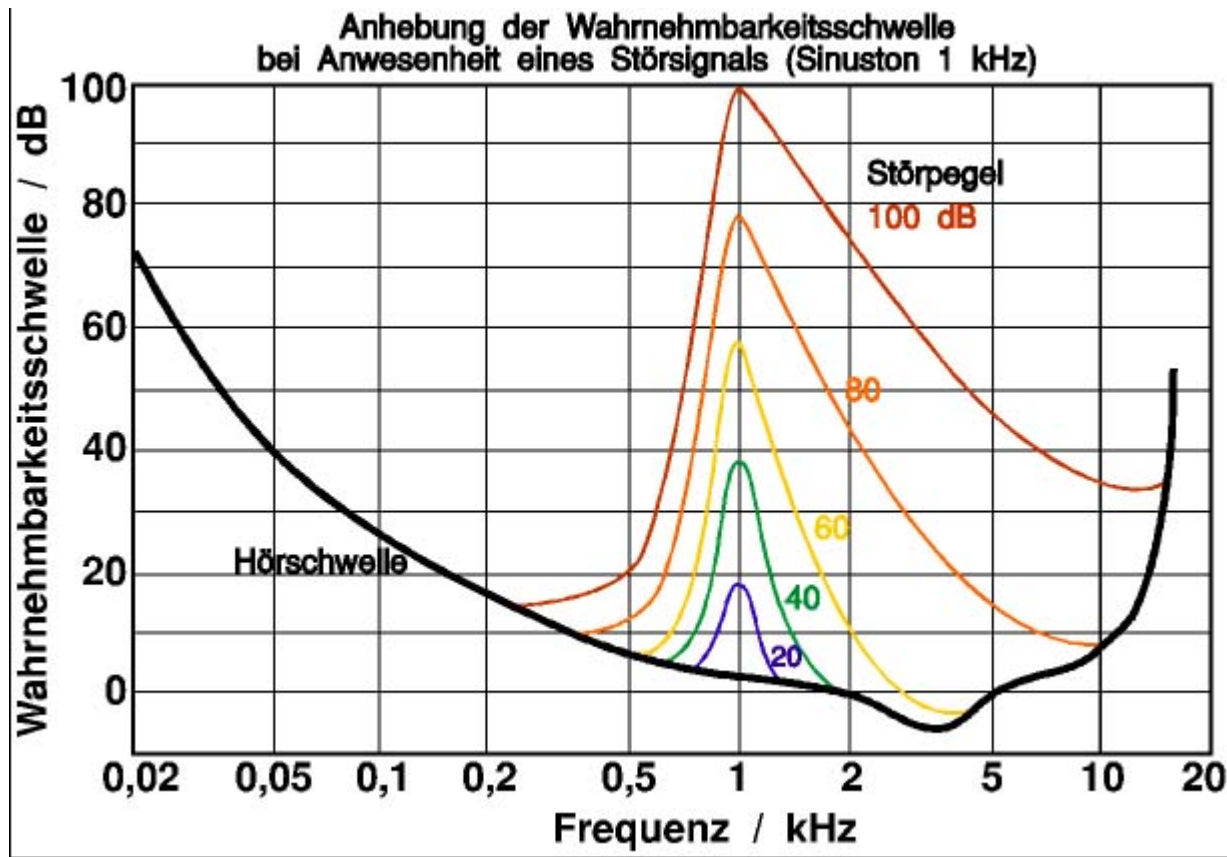
Maskierungseffekt

Maskierungseffekte (auch Verdeckung genannt) bewirken beim menschlichen Gehör, dass der Mensch in einem Geräusch bestimmte Frequenzanteile nicht oder nur mit verringerter Sensitivität wahrnehmen kann.

So ist das Gehör beispielsweise nicht in der Lage, bei sehr lauten Bässen gleichzeitig sehr leise Töne im mittleren Frequenzbereich wahrzunehmen. Die Bässe maskieren hier die Mitten. Der Mindestpegel, von dem an diese Mitten wahrgenommen werden, hängt in diesem Beispiel vom Pegel des Basssignals und vom Frequenzabstand zwischen Basston und Mittenton ab.

Im Bild ist die Wirkungsweise von Maskierungseffekten dargestellt. Ist zum Beispiel ein 1-kHz-Ton mit

einem Schallpegel von 80 dB anwesend, so kann ein 2-kHz-Ton von 40 dB nicht mehr wahrgenommen werden. Das heißt, der 2-kHz-Ton kann weggelassen werden, ohne dass ein Mensch diesen Unterschied hört. Tritt zusammen mit einem 1-kHz-Ton von 80 dB ein 2-kHz-Ton von 60 dB auf, kann man beide Töne wahrnehmen. Aber man kann diesen 2-kHz-Ton mit sehr schlechter Qualität übertragen: Selbst Störgeräusche von 40 dB können vom Menschen nicht mehr wahrgenommen werden.



Bei Verfahren zur verlustbehafteten Audiodatenkompression, wie beispielsweise MP3 oder Ogg Vorbis, werden solche Maskierungseffekte gezielt ausgenutzt, um Frequenzanteile, die wegen Maskierung zur Zeit unhörbar sind, für diesen Moment auszufiltern, oder um Frequenzbereiche, die teilweise maskiert werden, mit geringerer Qualität (d. h. mit geringerer Datenrate) zu übertragen.

Für die Messung der wahrgenommenen Lautstärke spielen die Maskierungseffekte eine wesentliche Rolle. Denn hierüber wird beschrieben, welche Nervenzellen von einem Geräusch überhaupt angeregt werden. Die Summe aller Nervenerregungen spiegelt die empfundene Lautstärke wider.

Hochpassfilter

Ein Hochpassfilter lässt, wie der Name erraten lässt, Audiosignale oberhalb einer bestimmten (möglicherweise einstellbaren) Frequenz durch, während Frequenzen darunter abgedämpft werden.

Tiefpassfilter

Ein Hochpassfilter lässt, wie der Name erraten lässt, Audiosignale unterhalb einer bestimmten (möglicherweise einstellbaren) Frequenz durch, während Frequenzen darüber abgedämpft werden.

Digitalisierung von Audio

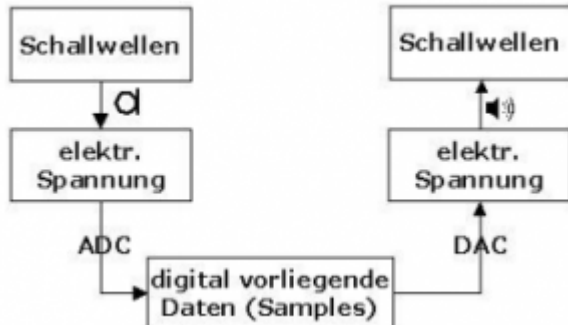
Grundsätzlich sind Daten im Audibereich zeitkritisch, d.h. sie stellen Anforderungen an Hard- und Software, die historisch gesehen atypisch für einen 'Rechner' sind. Bis heute sind damit Geschwindigkeits-, Bandbreiten- und Speicherprobleme verbunden, die von technischen und ökonomischen Rahmenbedingungen abhängen.

Um den Sound am Computer bearbeiten zu können, müssen die analogen elektrischen Signale in digitale Daten umgewandelt werden.

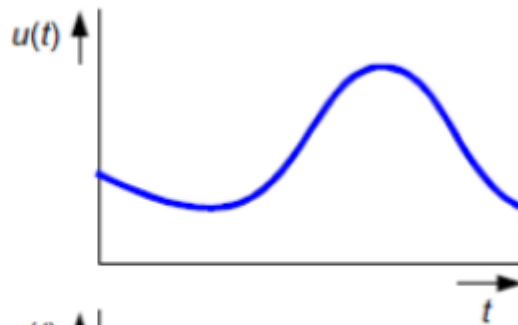
Damit Audiodaten aufgenommen und wiedergegeben werden können, werden **A(nalog)/D(igital)-Wandler (AD-Converter)** bzw. **D/A-Wandler (DA-Converter)** benötigt. Diese werden i.d.Regel von der Soundkarte zur Verfügung gestellt.

- hierbei wird die analoge Spannung in einem regelmäßigen Zeittakt (**Samplingfrequenz**) gemessen.
- Diese Messwerte sind je nach ausgewählter **Bittiefe** aufgelöst
- heraus kommt eine zahlencodierte Nachbildung des analogen Signals als eine **Abfolge von diskreten Schritten**

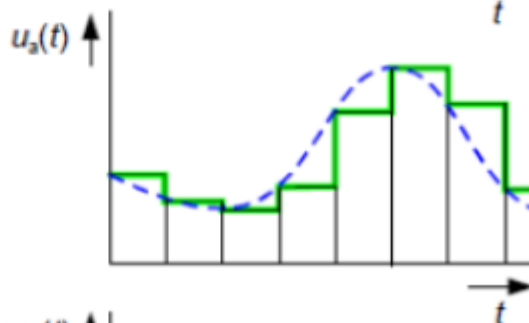
Beim Abspielen dieser Meßwerte wandelt ein Digital-AnalogConverter (DAC) den Zahlencode wieder in eine Spannung um, die vom Lautsprecher wieder in Schalldruck umgesetzt wird.



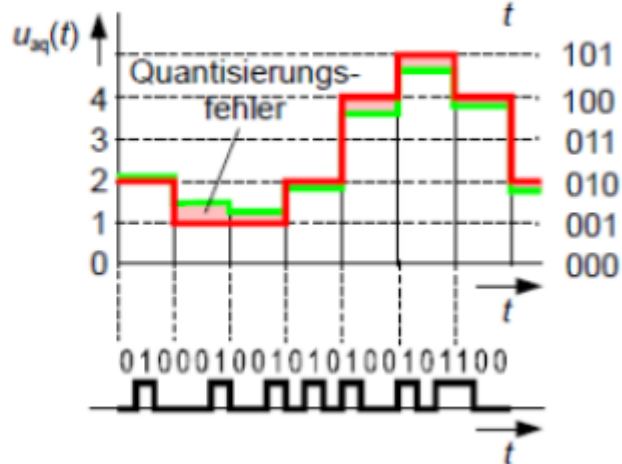
Beim Computer befinden sich die beiden Converter in der Soundkarte oder sind schon direkt im Mainboard integriert.

Kontinuierliches Signal:**Abgetastetes Signal:**

(S&H 100%)

**Quantisiertes Signal:**

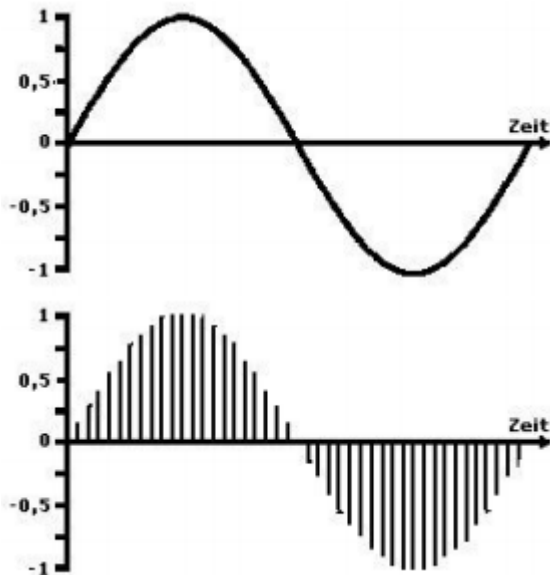
(lineare Quantisierung)

**Digitalsignal:****Kanäle**

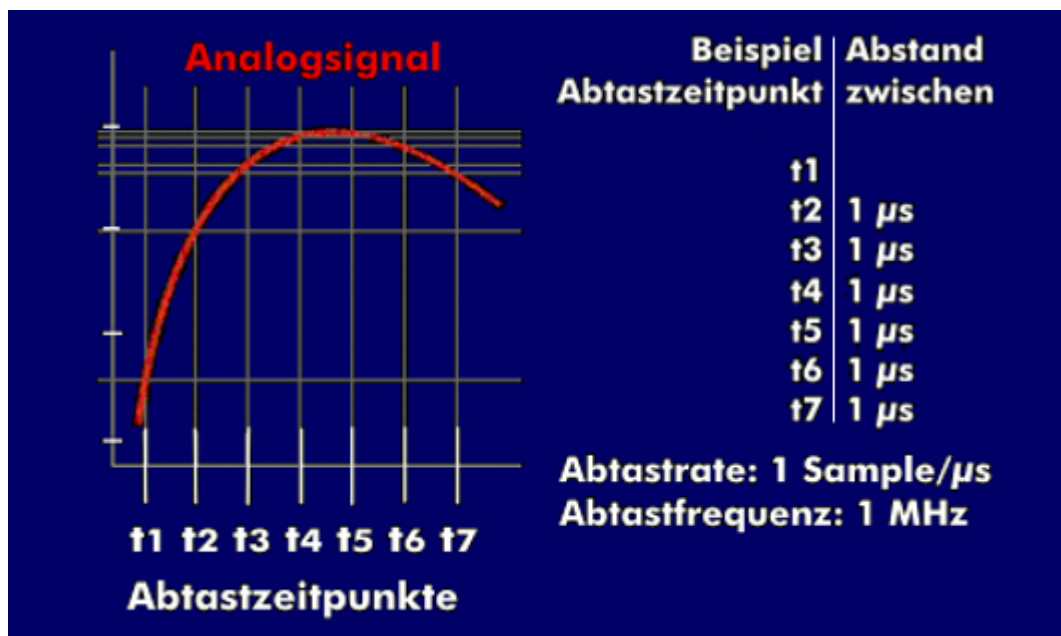
Eine Audiodatei kann mehrere Kanäle haben. Verbreitet sind Mono (ein Kanal), Stereo (2 Kanäle) und 5.1 bzw. 7.1 (Surround). Ein Kanal liefert jeweils die Informationen für einen der Lautsprecher und ist ein eigenständiges Audiosignal. Das heißt wir können eine Stereo-Datei aufteilen und in zwei Mono Dateien speichern.

Samplingrate/Samplingfrequenz

Bei der Digitalisierung werden in **regelmäßigen zeitlichen Abständen Messproben (Samples)** des analogen Ausgangssignals entnommen. Die **Abtastrate** wird **pro Sekunde** angegeben.



z.B. :Für die Audio-CD wird mit 44.100 Hz (also 44100 mal in der Sekunde) gemessen.



Aliasing

Beim Digitalisieren können Frequenzen entstehen, die im Original nicht vorhanden sind. Diese Artefakte oder 'Aliasfrequenzen' des Digitalisierungsvorgangs sind Ergebnisse einer **unzureichenden Abtastfrequenz**. Die **Abtastrate** muss **mindestens das Doppelte der zu digitalisierenden Frequenz** (siehe Abtasttheorem nach Nyquist/Shannon) betragen. Je höher die Abtastrate, desto besser ist die Audioqualität nach der Digitalisierung.

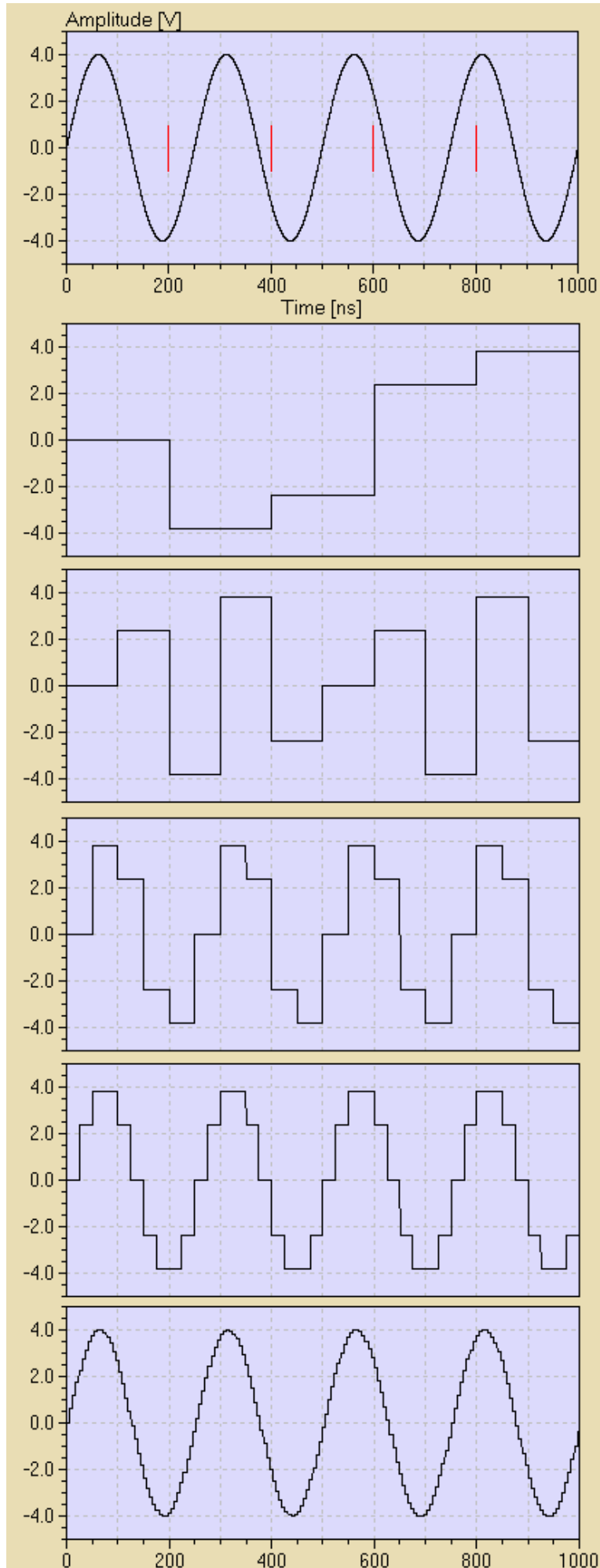
Beispiel

Das folgende Originalsignal zeigt in der Skizze eine Sinuswelle mit der Schwingungsdauer von $T=250\text{ns}$.

Das heißt die Frequenz ($1/s = \text{Hz}$) des Originalsignals ist:

$$1/250\text{ns} \Rightarrow 1/250 \cdot 10^{-9} \text{ s} \Rightarrow 1/250 \cdot 10^{-9} \text{ s} \Rightarrow 4000000 \cdot 1/\text{s} \Rightarrow 4000000 \text{ Hz} \Rightarrow 4 \text{ MHz}$$

D.h. Um ein 4-MHz-Signal korrekt wiedergeben zu können ist eine minimale Abtastrate von 8 MHz notwendig. Wird diese Grenzfrequenz unterschritten, entstehen Artefakte, die mit dem ursprünglichen Signal nichts mehr zu tun haben.



Original-signal

Abtastrate
5 MHz

Abtastrate
10 MHz

Abtastrate
20 MHz

Abtastrate
40 MHz

Abtastrate
160 MHz

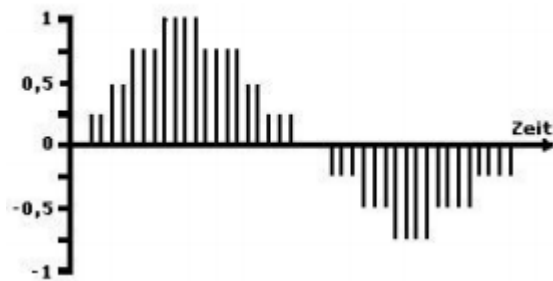
Bittiefe

Für die jeweiligen Samples stehen nur eine begrenzte Anzahl an Messwerten zur Verfügung. Die für das Digitalisieren gewählte Bittiefe bestimmt diese Menge, die sich über den Messbereich verteilt.

Mit der **Bittiefe ist die Länge der Binärzahlen** gemeint, aus denen die digitale Audiodatei letztendlich besteht. Binärzahlen sind Zahlen, die nur aus Nullen und Einsen bestehen. Ein Computer rechnet in seinem Inneren nur mit diesen Zahlen.

- 1 Bit $\rightarrow 2^1 = 2$ Zustände (0 oder 1)
- 2 Bit $\rightarrow 2^2 = 4$ Zustände (0 bis 3)
- 3 Bit $\rightarrow 2^3 = 8$ Zustände (0 bis 7)
- ...

Abtastung mit Bittiefe = 3



Bitrate

Die Bitrate (kBit/s) wird gern mit der Auflösung verwechselt. Sie steht für die „Bandbreite“ der Audiodatei, also welche Datenmenge in einer Sekunde verarbeitet wird. Für unkomprimierte Formate wie WAV und AIFF berechnet man die Bitrate ganz einfach, indem man die drei Werte von oben multipliziert:

$$\text{Bitrate} = \text{Kanäle} \times \text{Samplingrate} \times \text{Bittiefe}$$

Beispiel

Eine Stereo WAV-Datei in CD-Qualität hat eine Samplingrate von 44,1kHz & Bittiefe von 16Bit.

$$2 \text{ Kanäle} \times 16 \text{ Bit} \times 44,1 \text{ kHz} = 1411,2 \text{ kBit/s}$$

unkomprimierte Audioformate

Für unkomprimierte Audiosignale gibt es für den Computer die Formate **WAV(Windows)**, **AIFF (Macintosh)**, **AU (Linux)** und **PCM**.

Diese Formate unterscheiden sich nicht wesentlich. Sie können Audio prinzipiell in jeder Samplingfrequenz und Bittiefe speichern. Die Wahl richtet sich dabei nach der gewünschten Wiedergabequalität, der Downloadgeschwindigkeit und der Speicherplatzverfügbarkeit.

Üblich sind folgende Kombinationen:

Qualität	Bittiefe	Samplingfrequenz Hz	Anzahl Kanäle	Downloadrate kbit/s	Speicherbedarf MByte/min
Telefon	8	8000	1 (Mono)	62,5	0,5
Radio	16	22050	1 (Mono)	352,8	2,8
CD	16	44100	2 (Stereo)	1378,1	11
DVD	24	48000	6 (5.1-Sound)	6750,0	54

komprimierte Audioformate

Für die meisten Anwendungen ist der Speicherverbrauch unkomprimierter Formate zu hoch (z.B. für Downloads aus dem Internet, als Soundtrack für Videos auf CD oder DVD)

Für solche Fälle gibt es Formate, welche die Datenmenge reduzieren, indem sie nicht hörbare Frequenzen eliminieren.

Diese Verfahren verschlechtern die Qualität mit jedem Kompressions- und Dekompressionsvorgang und sind daher für die Zwischenspeicherung nicht zu empfehlen. Es sollte daher beim Editieren möglichst lange mit der unkomprimierten Datei gearbeitet werden, bis man das Endresultat schließlich für die finale Anwendung passend komprimiert.

Beispiele für die Anwendung vom Komprimierungs-Codecs:

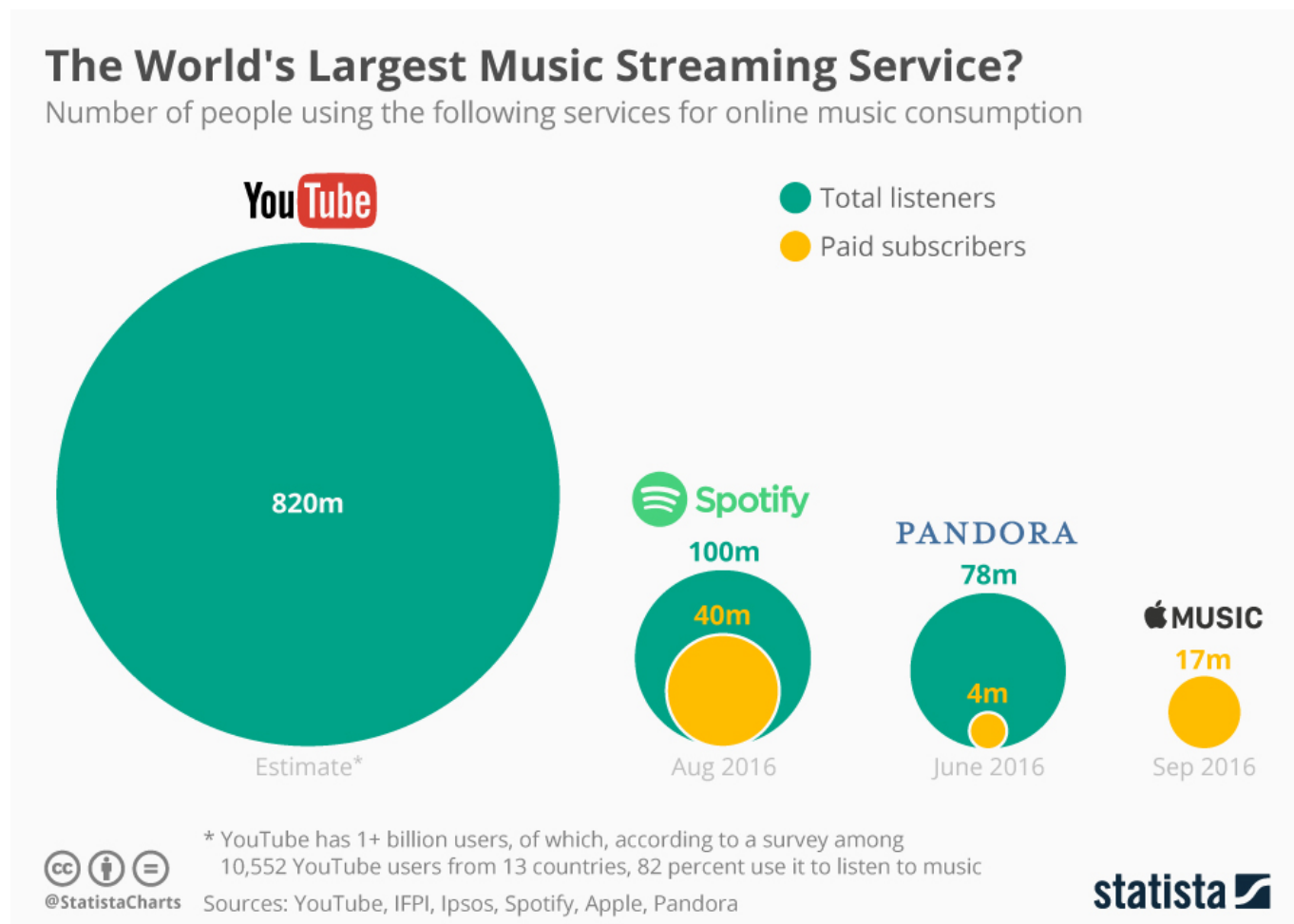
Anwendung	Qualität	Codecs	Original kbit/s	Komprimiert kbit/s	Kompressionsfaktor
Internet Echtzeit-Streaming	Telefon bis Radio	mp3, mpeg4, wma, RealAudio, ogg, vorbis	689-1378	30-60	ca. 11-45
Internet Download	quasi-CD	mp3, wma, ogg, vorbis	1378	128	10,7
DVD	quasi-DVD	AC3, mpeg2	6750	448 (AC3)	15(AC3)

Streamingdienste

Streaming Audio ist eine Variante des Streaming Media, bei der Audiodaten kontinuierlich über ein Rechnernetz übertragen werden. In seiner einfachsten Form kann man sich Streaming Audio als Hörfunk im Internet vorstellen.

Musik-Streaming beschreibt die Option, jederzeit stationär oder mobil auf viele Millionen Musikstücke zugreifen zu können. Die Songs und Alben lagern in der Cloud und können via Internetverbindung bei verschiedenen Anbietern auf Anfrage abgerufen werden. Das Prinzip des Musikhörens auf Abfrage per Flatrate liegt voll im Trend. Eindeutige Indizien liefern stetig steigende Nutzerzahlen der Abo-Dienste. Die wichtigsten Anbieter der Branche verzeichnen ein konstantes bis atemberaubendes Wachstum. Ein Ende der Erfolgsgeschichte ist vorerst nicht abzusehen.

Spotify nennt diesbezüglich mittlerweile Zahlen weit jenseits der 100 Millionen-Marke. Auch Apple Music ist auf dem Vormarsch und spricht von knapp 30 Millionen zahlenden Nutzern. Immer mehr Menschen setzen auf Musik-Streaming und lassen die CD oder auch Schallplatte im Regal verstauben. Doch die Konkurrenz schläft nicht. Der Markt hält zahlreiche weitere Angebote bereit. So buhlen unter anderem Google mit Play Music, Amazon mit Prime Music und Music Unlimited, Napster, Deezer, Tidal oder auch SoundCloud Go um die Gunst der Hörerschaft.

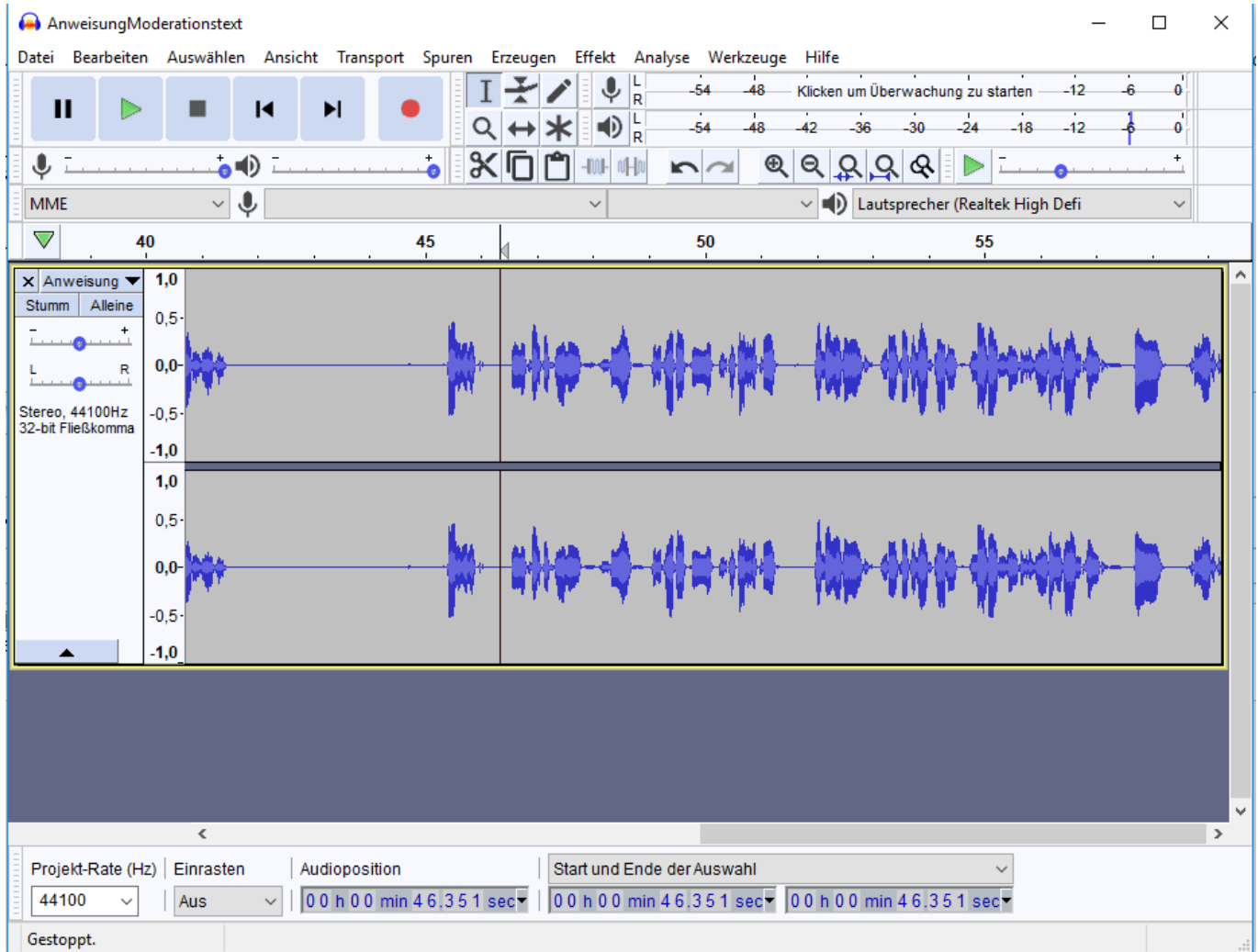


8.2) Praxis mit Audacity

Die Audiotbearbeitung gehört zu den Grundlagen der Musikproduktion und wird auch »Audio Editing« genannt. Die Bearbeitung von Audiodateien findet in der Regel vor dem Abmischen statt. Allerdings spricht man auch von Audiotbearbeitung, wenn man nur Knackser aus MP3 Dateien entfernen oder MP3s schneiden möchte.

Ein mögliches Programm für Audio Editing ist:

- [Programm Audacity](#)
- [Eine Anleitung auf YouTube](#)
- [Eine Reihe kurzer Filme zur Einführung in Audacity](#) -> **mit Microsoft Edge öffnen**
- [Shortcuts Audacity](#)
- [Übungen](#)



Destruktive vs. Non-destruktive Audiotbearbeitung

Sehr wichtig für deine Audiodateien ist die Unterscheidung zwischen **destruktiver** und **non-destruktiver Audiotbearbeitung**. Als **destruktiv** wird die Bearbeitung von Audiodateien dann bezeichnet, wenn die **Änderung an der Originaldatei** selbst durchgeführt und festgeschrieben werden (Bsp. Audacity). Ist die Änderung erst einmal gemacht, lässt sie sich **nicht mehr rückgängig** machen.

Bei der **non-destruktiven Audiotbearbeitung** werden die Änderungen **nicht fest in die Datei geschrieben** (Bsp. Reaper). Das Programm zur Bearbeitung merkt sich in diesem Fall nur, wie die Datei zu klingen hat und **berechnet die Änderungen in Echtzeit beim Abhören**. Möchtest Du die **Änderungen festschreiben**, so musst Du die Datei **exportieren (bouncen)** – in eine neue Datei. Dein **Original bleibt jederzeit unangetastet** und Du kannst schnell zum Ursprung zurückkehren.

Vorteile non-destruktiv

- Das Original ist jederzeit sicher
- Du kannst immer wieder von vorne beginnen ...
- und ganze neue Effekte ausprobieren
- Fehler lassen sich leicht ausmerzen

Vorteile destruktiv

Selbst wenn die Audiorechner heute eine Menge Leistung bereithalten – sie ist endlich. Und spätestens hier kommt die destruktive Audiotbearbeitung ins Spiel. Schreibst Du die Effekte in die Datei fest (destruktiv), wird keine zusätzliche Rechenleistung für die Berechnung der Effekte mehr benötigt. Diese Audiodatei wird von allen Geräten und Musikprogrammen in exakt demselben Zustand abgespielt.

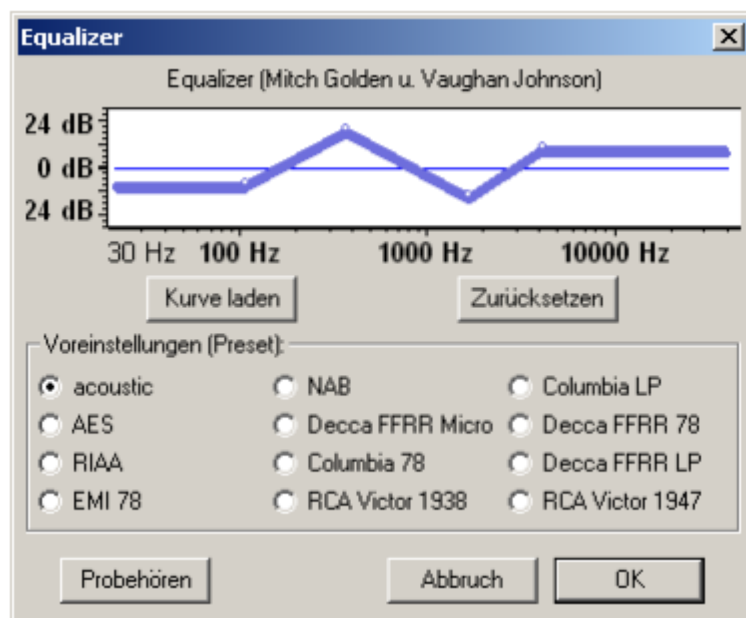
Normalisieren von Audio

Als **Normalisierung** wird in der Audiotechnik der Vorgang bezeichnet, die Amplitude analogen oder digitalen Audiomaterials so zu vergrößern oder zu verkleinern, dass sie innerhalb eines vorgegebenen Bereichs liegt. Es gibt im Wesentlichen zwei Gründe, warum normalisiert wird – zum einen technische Rahmenbedingungen bei der Übertragung bzw. Speicherung des Audiomaterials, zum anderen die Anpassung der Lautheit (vom Menschen empfundene Lautstärke) des Materials.

Die Funktion „Normalisieren“ ist zum Beispiel dann sinnvoll, wenn **mehrere Musikstücke hintereinander** abgespielt werden sollen. Mit der Normalisierung der Audiodaten kann eine gleichmäßigere Lautstärke über das gesamte Album hinweg erreicht werden bzw. verhindert werden, dass einzelne Stücke so leise im Vergleich sind, dass der Hörer die Lautstärke an der Stereoanlage verändern muss.

Equalizer

Erlaubt einzelne Bereiche des Frequenzspektrums gezielt lauter oder leiser zu machen.



Effekt: Fade-In & Fade-Out (Einblenden & Ausblenden)

Fade-Ins und Fade-Outs werden dazu genutzt, das Audiosignal **sanft ein- bzw. auszublenden**. Das kennst Du beispielsweise aus dem Radio, wo ein Song reinkommt oder wieder rausgefahren wird,

während die Moderation startet.

Wenn bei einer Aufnahme die Welle nicht mit einem Nullpunkt beginnt, führt dies zu einem mehr oder minder leisen Knackser. Auch an den Übergängen zweier Audioclips kann es zu einem solchen Knacksen kommen. Eine sehr kurze Einblendung, ein Fade-In zu Beginn und eine Ausblendung, ein Fade-Out am Ende sorgen hier für Nullpunkte und damit knackserfreie Übergänge.

Viele Musik- und Audiotbearbeitungsprogramme machen das inzwischen automatisch am Beginn und am Ende einer Audiodatei. Die Handbücher weisen dieses Feature als »automatic fade«, »Auto-Fade«, »automatische Ein-/Ausblendung« oder dergleichen aus.

Tempo ändern

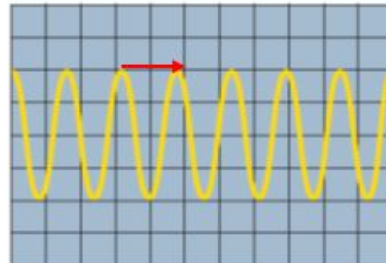
Das Tempo, auch Zeitmaß, gibt in der Musik an, wie schnell ein Stück zu spielen ist, bestimmt also die absolute Dauer der Notenwerte.

Tonhöhe ändern

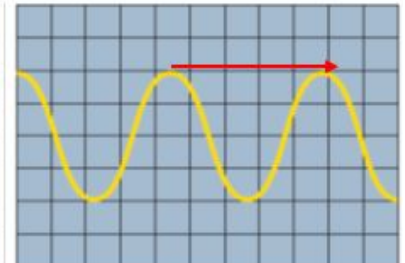
Die Tonhöhe wird in der Praxis mit der **Frequenz (Schwingungsanzahl pro Zeitspanne) eines hörbaren Tones** gleichgesetzt, genauer: mit der **Grundfrequenz**, also der Frequenz des tiefsten Teiltons in dem betreffenden Klang. Die Tonhöhe ist neben der Tondauer, der Lautheit und der Klangfarbe eine wichtige Eigenschaft musikalischer Töne und der Vokale gesprochener Sprache.

Die Höhe eines Tons ist von der Schwingungshäufigkeit (Frequenz) abhängig.

Einheit: Hertz (Hz)
1 Hertz bedeutet 1 Schwingung pro Sekunde.



hoher Ton



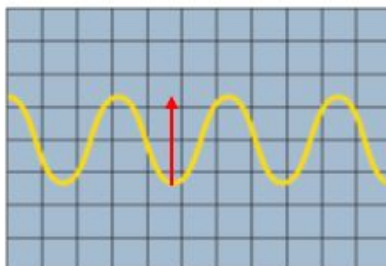
tiefer Ton

Lautstärke ändern

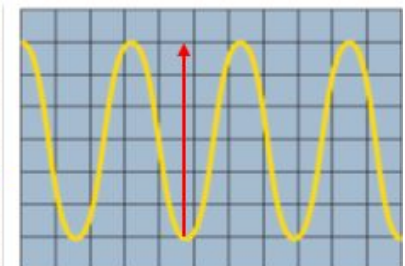
Die Lautstärke ist die physikalisch messbare Amplitude oder Stärke des Schalls (z. B. als Schalldruck bzw. als Schalldruckpegel).

Die Lautstärke eines Tons hängt von der Schwingungsweite (Amplitude) ab.

Einheit: Dezibel (Db)
Dezibel ist keine lineare Einheit – mit ihrer Hilfe kann man Lautstärken vergleichen.



leiser Ton



lauter Ton

Audioformate

Die Vielzahl der digitalen Audio ist für den Nutzer manchmal verwirrend. Hier findest du eine Übersicht der gängigsten Audioformate und deren Dateiendungen inklusive einer Kurzbeschreibung.

.WAV - wave form audio

Das WAV Dateiformat dient der digitalen Speicherung von Audiodaten. In den allermeisten Fällen enthält eine WAV Datei dabei unkomprimierte Rohdaten. Der Verzicht auf eine Komprimierung bedeutet folglich, dass die Audiodatei in höchster Qualität gespeichert ist. Allerdings erfordert dies einen enormen Speicherbedarf. Zwei Minuten Musik können schnell um die 20 Megabyte (MB) Platz belegen.

.MP3 - MPEG-1 Audio Layer 3

Von einer Expertengruppe entwickeltes Audioformat mit standardisierten Komprimierungsverfahren. Die Vorteile von MP3 Audiodateien sind die vergleichsweise kleinen Dateien bei dennoch sehr guter Qualität. In der Praxis bedeutet dies, dass Sie den Inhalt einer kompletten Musik CD ohne wahrnehmbare Qualitätsverluste auf die Größe von 10 bis 20 Megabyte (MB) reduzieren können.

.WMA - Windows Media Audio

Ein von Microsoft © entwickeltes Audioformat für Windows Betriebssysteme. Hohe Kompression mit guter Qualität. Das bedeutet relativ kleine Dateien und dennoch Musik- oder Sounddateien mit guter Qualität möglich.

.AAC - Advanced Audio Coding

Ein ebenfalls von der MPEG (Moving Picture Experts Group) entwickeltes Audiodatenkompressionsverfahren. AAC gilt als qualitativ verbesserter Nachfolger von MP3. Die Vorteile von AAC Audiodateien sind die vergleichsweise kleinen Dateien bei dennoch exzellenter Qualität. In der Praxis bedeutet dies, dass Sie den Inhalt einer kompletten Musik CD ohne wahrnehmbare Qualitätsverluste auf die Größe von 10 bis 20 Megabyte (MB) reduzieren können.

OGG - Ogg Vorbis genannt

Ein freier (patentfreier) Codec zur verlustbehafteten Audiodatenkompression, der von der Xiph.Org Foundation als Alternative zum weit verbreiteten MP3 Format entwickelt wurde.

Creative Commons Lizenzen

Creative Commons (abgekürzt CC; englisch für schöpferisches Gemeingut, Kreativallmende) ist eine

gemeinnützige Organisation, die 2001 in den USA gegründet wurde. Sie veröffentlicht verschiedene **Standard-Lizenzverträge**, mit denen ein **Autor der Öffentlichkeit** auf einfache Weise **Nutzungsrechte an seinen Werken einräumen** kann. Diese Lizenzen sind nicht auf einen einzelnen Werkstyp zugeschnitten, sondern für beliebige Werke anwendbar, die unter das Urheberrecht fallen, zum Beispiel Texte, Bilder, Musikstücke, Videoclips usw. Auf diese Weise entstehen Freie Inhalte.

Entgegen einem häufigen Missverständnis ist Creative Commons nicht der Name einer einzigen Lizenz, sondern einer Organisation. Die **verschiedenen Lizenzen von Creative Commons** unterscheiden sich erheblich. Einige CC-Lizenzen schränken die Nutzung relativ stark ein, andere wiederum sorgen dafür, dass auf das Urheberrecht so weit wie möglich verzichtet wird. Veröffentlicht jemand beispielsweise ein Werk unter der Lizenz CC-BY-SA, dann erlaubt er die Nutzung durch andere Menschen unter der Bedingung, dass der Urheber sowie die betreffende Lizenz angegeben werden. Darüber hinaus darf der Nutzer das Werk unter der Bedingung verändern, dass er das bearbeitete Werk unter derselben Lizenz veröffentlicht

[siehe Creative Commons Lizenzen - WICHTIG, da prüfungsrelevant!](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:1

Last update: **2019/11/19 16:52**



Shortcuts in Audacity

Leertaste	Abspielen - Play/Stop
F1	Auswahlwerkzeug
F2	Hüllkurvenwerkzeug - Lautstärke verstellen
F3	Zeichenwerkzeug - einzelne Samples verändern
F4	Zoomwerkzeug / ev. mit Shift
Strg+Mausrad	Zoomen
F5	Bewegen von Auswahlbereichen
Strg+d	Duplizieren des Auswahlbereichs auf neue Spur
Strg+x	Ausschneiden
Strg+Alt+x	Ausschneiden, Spur rückt nicht nach
Strg+v	Einfügen
Strg+t	Nur Auswahlbereich übrig lassen
Entf	Entfernen
Strg+Alt+k	Löschen, Spur rückt nicht nach
Strg+l	in Stille umwandeln
Strg+i	Trennen

Rauschentfernung

- Effekt - Rauschentfernung
 - zuerst Rauschprofil ermitteln (Teil, wo Rauschen auftritt)
 - Rauschentfernung

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:1:shortcuts_audacity

Last update: **2019/09/08 10:41**



Übungen

- 1) Übung Sprechpuppe
- 2) Übung Audioschnitt
- 3) Übung Störquelle
- 4) Übung Audibearbeitung
- 5) Übung Samplingrate

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:1:uebungen



Last update: **2019/09/09 23:08**

Übung 3: Störquelle

- a) Hören Sie sich die Datei „dm0809_ue5.wav“ in Audacity an. Beschreiben Sie die Störquelle, welche die Stimme überlagert.
- b) Mit welchem Filter könnte man diese Störung beheben und warum? Beschreiben Sie den Filter kurz.
- c) Versuchen Sie, die störende Frequenz aus der Audiodatei zu entfernen. Nutzen Sie dafür die Frequenzanalyse (Menü: Analyse → Spektrum) und den Filter, für welchen Sie sich in Aufgabe 2b) entschieden haben. Die Filter finden sie im Menü unter „Effekt → Filtername“ (Audacity 1.2) oder dem entsprechenden Unterverzeichnis (Audacity 1.3). Beachten Sie dabei, dass Bereiche, auf die eine Analyse oder ein Effekt angewendet sollen markiert sein müssen. Außerdem können Effekte wiederholt werden, um das Ergebnis zu verbessern. Beschreiben Sie, wie Sie vorgegangen sind (welche Analysemethoden und Filter haben Sie benutzt, in welcher Reihenfolge und mit welchen Einstellungen).
- d) Exportieren Sie die fertig bearbeitete Datei im WAV-Format (Menü: Datei → Exportieren als WAV ...) und legen Sie sie Ihrer Abgabe bei.

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:1:uebungen:3

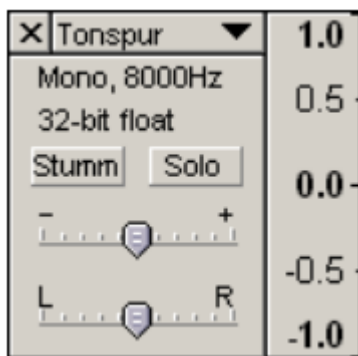
Last update: **2019/09/08 14:06**



Übung 4: Audioaufnahme

a) Erstellen Sie ein neues Projekt in Audacity. Schließen Sie ein Mikrofon an Ihren Rechner an oder benutzen das interne und nehmen Sie mittels des Aufnahme-Buttons Ihren Namen auf.

b) Die Aufnahme erscheint in einer neuen Tonspur. Wenn Sie auf den Pfeil neben „Tonspur“ klicken, öffnet sich das Menü für diese Tonspur. Unter „SAMPLERRATE EINSTELLEN“ können Sie die Werte der Samplerrate anpassen. Nutzen Sie diese Funktionalität, um einen „Chipmunk-Effekt“ zu erzeugen. D.h. Ihre Stimme soll sich höher anhören. Erklären Sie den Zusammenhang zwischen der veränderten Samplerrate und diesem Effekt! Exportieren Sie die neu erstellte Datei im WAV-Format (Menü: Datei → Exportieren als WAV ...) und fügen damit automatisch alle Tonspuren zu einer zusammen. Nennen Sie die Datei „chipmunk.wav“ und legen Sie diese bei der Abgabe bei.



c) (optional) Stellen Sie die Samplerrate wieder auf den ursprünglichen Wert zurück. Fügen Sie ihrem Projekt nun mindestens zwei Störquellen hinzu. Nutzen Sie dazu die Möglichkeit neue Tonspuren über Projekt → Neue Tonspur einzufügen und Töne und Geräusche über die Einträge im Menü Generieren zu erzeugen. Beschreiben Sie alle Effekte und Einstellungen, die Sie benutzt haben. Exportieren Sie die neu erstellte Datei im WAV-Format (Menü: Datei → Exportieren als WAV ...) und fügen damit automatisch alle Tonspuren zu einer zusammen. Nennen Sie die Datei „damaged.wav“ und legen Sie sie Ihrer Abgabe bei.

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:1:uebungen:4

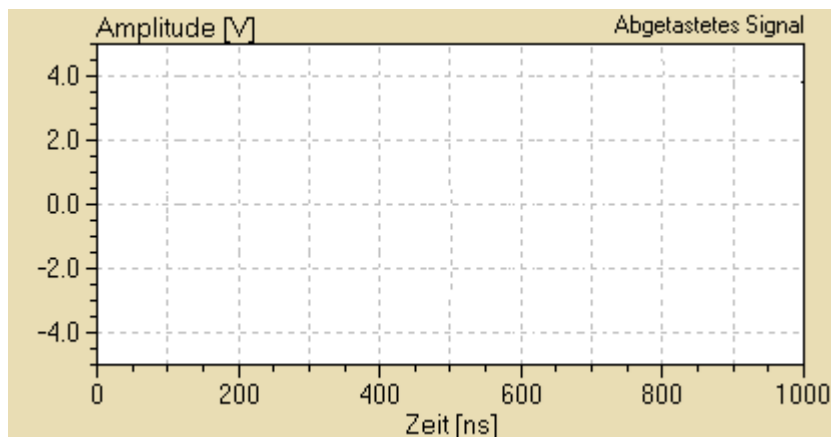
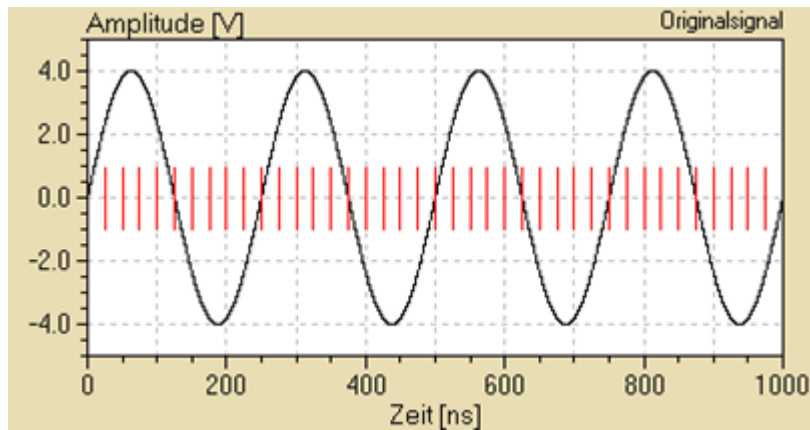
Last update: **2019/09/08 14:10**



Übung 5: Samplingrate

Erkläre die Begriffe Schwingungsdauer, Tonhöhe, Lautstärke, Samplingrate, Bittiefe & Aliasing in Zusammenhang mit Digitalisierung von Schallwellen.

Nutze folgende Abbildung für deine Erklärungen und zeichne ein abgetastetes Signal mit beliebiger Samplingrate ein.



From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:1:uebungen:5



Last update: **2019/09/09 23:08**

2) Videobearbeitung

Ein Video (vom lateinischen video = sehen) besteht aus einer **Abfolge von Einzelbildern in einer bestimmten Frequenz (Bildwiederholfrequenz)**. Diese werden grundsätzlich mit einer Optik und einem Bildwandler (Sensor oder Filmmaterial) erzeugt.

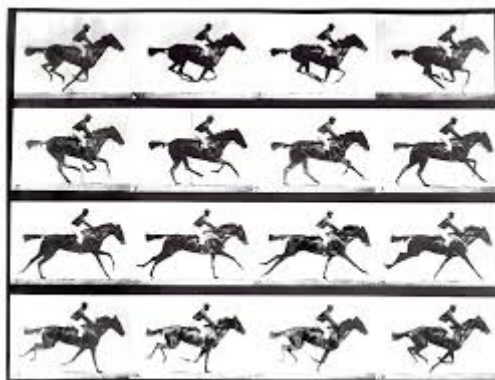
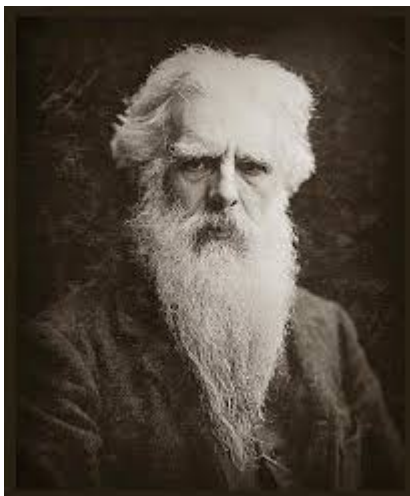
Es gelten **ähnliche fototechnische Prinzipien** (Blende, Belichtungszeit, Dynamik...) wie bei einer Spiegelreflex(DSLR/M)-Kamera. Zusätzlich wird jedoch der Datenstrom in ein bestimmtes Format **kodiert bzw. dekodiert (Codec)**, zB. MPEG oder h.264 und gespeichert. Als **Speichermedien** dienen Film, Magnetband, Solid-State-Speicher, Festplatten oder ähnliche Speicher mit entsprechenden Lese- und Schreibraten.

Bewegte Bilder

Beim Menschen entsteht eine **Illusion der Kontinuität ab ungefähr 12 Bildern pro Sekunde** (siehe zB. Daumenkino), falls die dargestellten Objekte langsam bewegt sind. Ab 16 Bildern pro Sekunde entsteht auch für schnellere Bewegungen ein solcher Eindruck. Der Psychologe Max Wertheimer und der Arzt/Forscher Peter Max Roget erkannten auf Basis der Gestaltpsychologie zwei dafür verantwortliche Phänomene:

- Die **Nachbildwirkung** (Trägheit der Retina) - Bsp. Laserpunkt
- Der **stroboskopische Effekt**, nach dem unser Gehirn Einzelbilder anders verarbeitet als eine Abfolge von Einzelbildern. Diese verschmelzen zu einem **Bewegungseindruck**, der von den Einzelbildern entkoppelt betrachtet werden kann.

Es entsteht somit eine zusätzliche Informationsdimension (für weitere Infos recherchiere den **Kuleshov-Effekt**). Das Gesamte ist somit auch in diesem Fall mehr als die Summe seiner Teile. Der Fotograf **Eadweard Muybridge (19. Jhdt.)** legte mit seinen Serienaufnahmen den Grundstein für die Entwicklung der Laufbildmedien. Sein **Zoopraxiskop** war der Vorläufer der Projektionstechnik in Kinos.



Aufgaben

A01) Recherchiere und beschreibe den **Kuleshov-Effekt**. Zeige weitere Dimensionen der Informationsvermittlung von Bewegtbildern (Schnitttechniken, Montage, ...)

A02) Recherchiere die Arbeiten von **Eadward Muybridge und der Anfänge der Chronofotografie**, vor allem sein **Zoopraxiskop** und seine sog. **Reihenfotografien**.

Videoformate

Wir unterscheiden grundsätzlich folgende Parameter der Videoformate:

- **Seitenverhältnis:** 4:3, 16:9, 21:9...
- **Bildpunkte bzw. Anzahl an Pixel gesamt und x/y:** 1920 px x 1080 px = 2 073 600 px
- **Bildwiederholfrequenz:** 25 Hz, 50 Hz...
- **Farbtiefe:** 8 bit, 10 bit...
- **Abtastung:** interlace (Zeilensprung) oder progressive (Progressiv)
- **Audio:** Stereo, Mehrkanal

In diesem Skript beschäftigen wir uns hauptsächlich mit dem **HDTV-Standard** und dem **SDTV-Standard**. UHDTV, 8K usw. können in den jeweiligen Spezifikationen der ITU (International Television Union), EBU (European Broadcast Union) oder SMPTE (Society of Motion Picture and Television Engineers) nachgelesen werden.

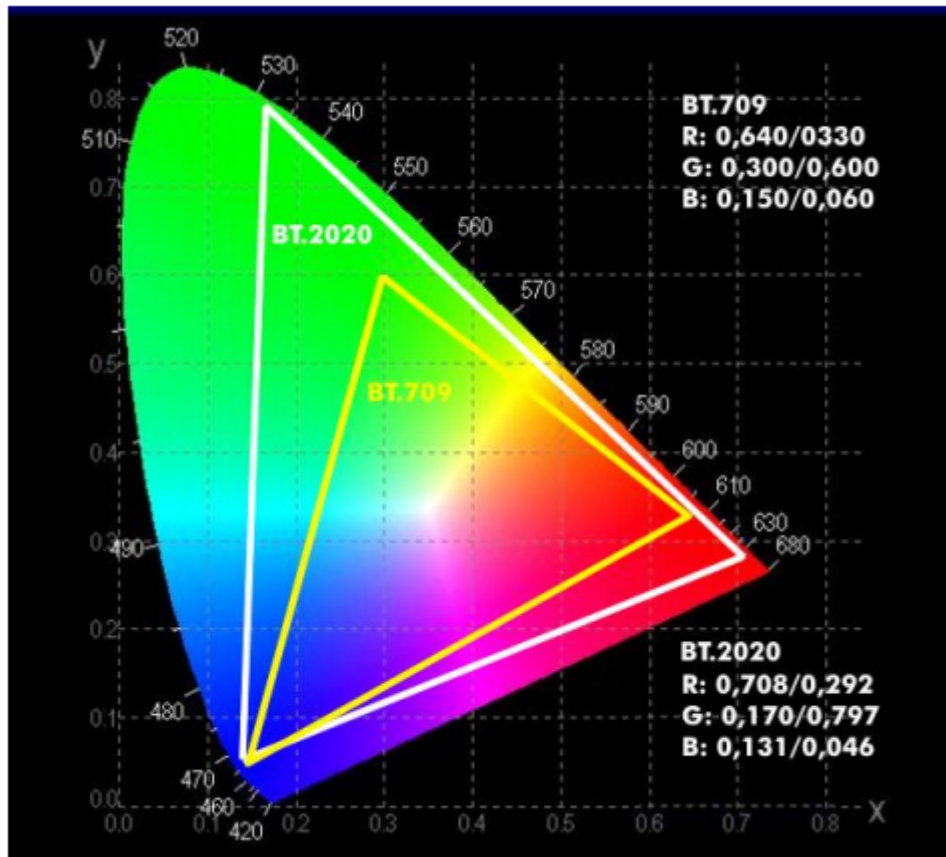
HDTV-Format

In Europa wird HDTV laut folgender [Spezifikation](#) definiert:

- **System 1** (S1) with **1280** horizontal samples and **720** active lines in **progressive** scan with a frame rate of **50 Hz**, **16:9** aspect ratio. Abbreviated 720p/50.
- **System 2** (S2) with **1920** horizontal samples and **1080** active lines in **interlaced** scan with a frame rate of **25 Hz**, **16:9** aspect ratio. Abbreviated 1080i/25.
- **System 3** (S3) with **1920** horizontal samples and **1080** active lines in **progressive** scan and a frame rate of **25 Hz**, **16:9** aspect ratio. Abbreviated 1080p/25.
- **System 4** (S4) with **1920** horizontal samples and **1080** active lines in **progressive** scan at a frame rate of **50 Hz**, **16:9** aspect ratio. Abbreviated 1080p/50.

Grundlage für diese Spezifikationen sind unter anderem die Empfehlungen der ITU, nämlich **ITU-R BT.709** und **ITU-R BT.2020**, welche die Regeln zur A/D-Wandlung eines analogen Komponentensignals beschreiben.

	BT.709	BT.2020
Bildpunkte	HD	UHD, 8K
Bildwiederholfrequenz	25, 24, 30, 50, 60	24, 25, 30, 50, 60, 100, 120
Zeilensprung/Progressiv	Beide	Nur progressiv
Farbraum	BT.709	BT.2020
Bittiefen	8, 10	10, 12
Farbmodell	RGB, YCbCr	RGB, YCbCr



Für die Übertragung der Signale im Produktionsumfeld wird HD-SDI verwendet, auf welches später eingegangen wird.

Common Image Format

Für **HD** wurde ein **gemeinsamer Standard** entwickelt, das **Common Image Format (CIF)**. Dabei ist das eigentliche Bildformat mit der Zeilen- und aktiven Bildpunktzahl bei allen Bildwechselfrequenzen gleich. Das Bildseitenverhältnis beträgt **16:9**. Die Bildpunkte sind quadratisch. Folgende grundsätzliche Normen gibt es:

- 1920 sichtbare Pixel x 1080 sichtbare Zeilen
- 1280 sichtbare Pixel x 720 sichtbare Zeilen

Man spricht in diesem Fall von der Anzahl der Nettozeilen und Nettopixel pro Zeile, die bei allen Bildwechselfrequenzen identisch sind. Dem sichtbaren Bild werden nicht sichtbare Pixel, die sogenannten Austastlücken („Blanking Intervalls“, in denen der Ton und Zusatzdaten übertragen werden), vorangestellt. Die Anzahl der nicht sichtbaren Pixel ist so gewählt, dass bei der digitalen Abtastung trotz unterschiedlicher Bildwechselfrequenzen (24, 25, 30, 50, 60 Hz) immer die gleiche Datenrate (von 1,485 GBit/s) erzeugt wird.

HD-Format	Nettozeilen	Bruttozeilen	Nettopixel / Zeile	Bruttopixel / Zeile
1080p/24	1080	1125	1920	2750
1080i/25	1080	1125	1920	2640
1080p/25	1080	1125	1920	2200
1080i/30	1080	1125	1920	1980
1080p/30	1080	1125	1920	
720p/50	720	750	1280	

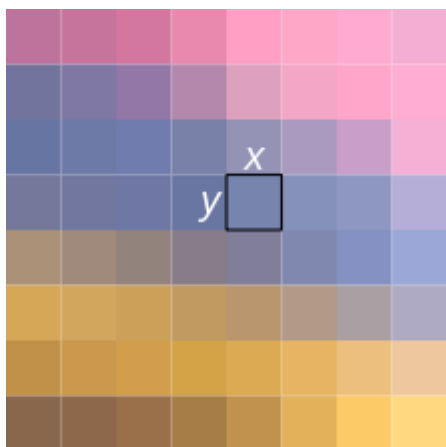
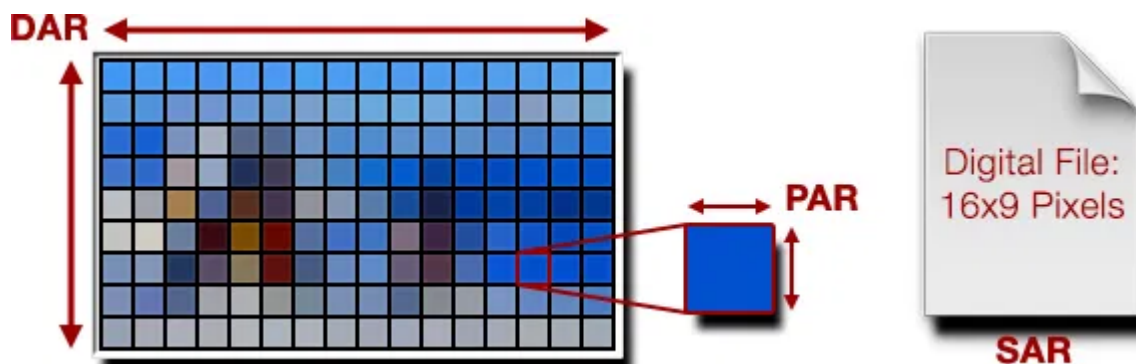
HD-Format	Bildfrequenz (Hz)	Auflösung sichtbar		Austattung		Gesamt	
		Horizontal (Pixel)	Vertikal (Pixel)	H-Blank (Pixel)	V-Blank (Pixel)	Horizontal (Pixel)	Vertikal (Pixel)
1080p/24	24	1920	1080	830	45	2750	1125
1080i/25	25 (2 Halbb.)	1920	1080	720	45	2640	1125
1080p/25	25	1920	1080	720	45	2640	1125
1080i/30	30 (2 Halbb.)	1920	1080	280	45	2200	1125
1080p/30	30	1920	1080	280	45	2200	1125
720p/50	50	1280	720	1980	30	1980	750

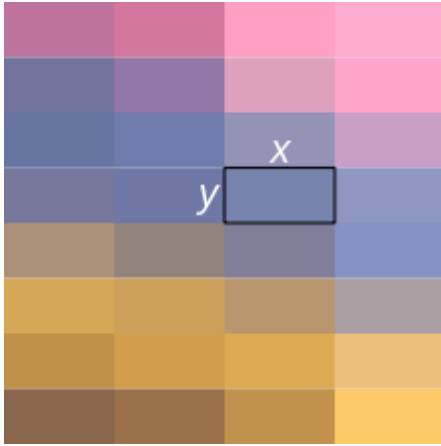
Berechnung der Bruttodatenraten anhand 1080i bei 10 bit:

$$1125 * 2640 * 2 * 10 * 25 = 1,485 \text{ Gbit/s}$$

Seitenverhältnis

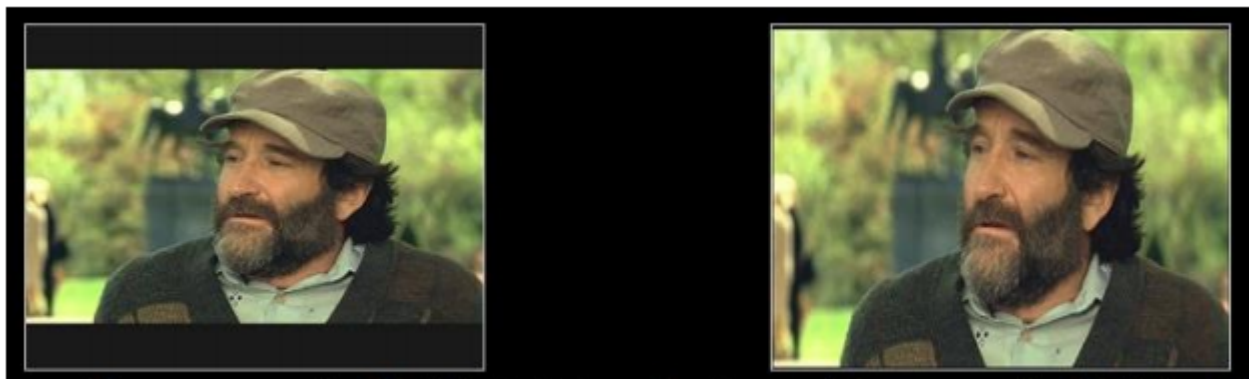
Aspect Ratio = Seitenverhältnis. Unterscheide SAR (Storage Aspect Ratio), DAR (Display Aspect Ratio) und PAR (Pixel Aspect Ratio). Wenn die Seitenverhältnisse für SAR und DAR übereinstimmen spricht man von **quadratischen Pixeln (square pixel)**, falls nicht, von **nicht-quadratischen Pixeln (nonsquare Pixel)**.





Storage-, Display- und Pixel-Aspect Ratio spielen dann eine Rolle, wenn zB. anamorph (verzerrt) aufgezeichnet wird wie bei DV-HDTV (1440 x 1080). Dabei wird in 4:3 anamorph aufgezeichnet (SAR) und bei der Wiedergabe (DAR) auf 16:9 entzerrt. Dabei ist das Pixel-Seitenverhältnis (PAR) 4:3, also nicht-quadratisch. Ein SAR von 1:1 entspricht dem DAR und somit quadratischen Pixeln

$$PAR = \frac{DAR}{SAR}$$

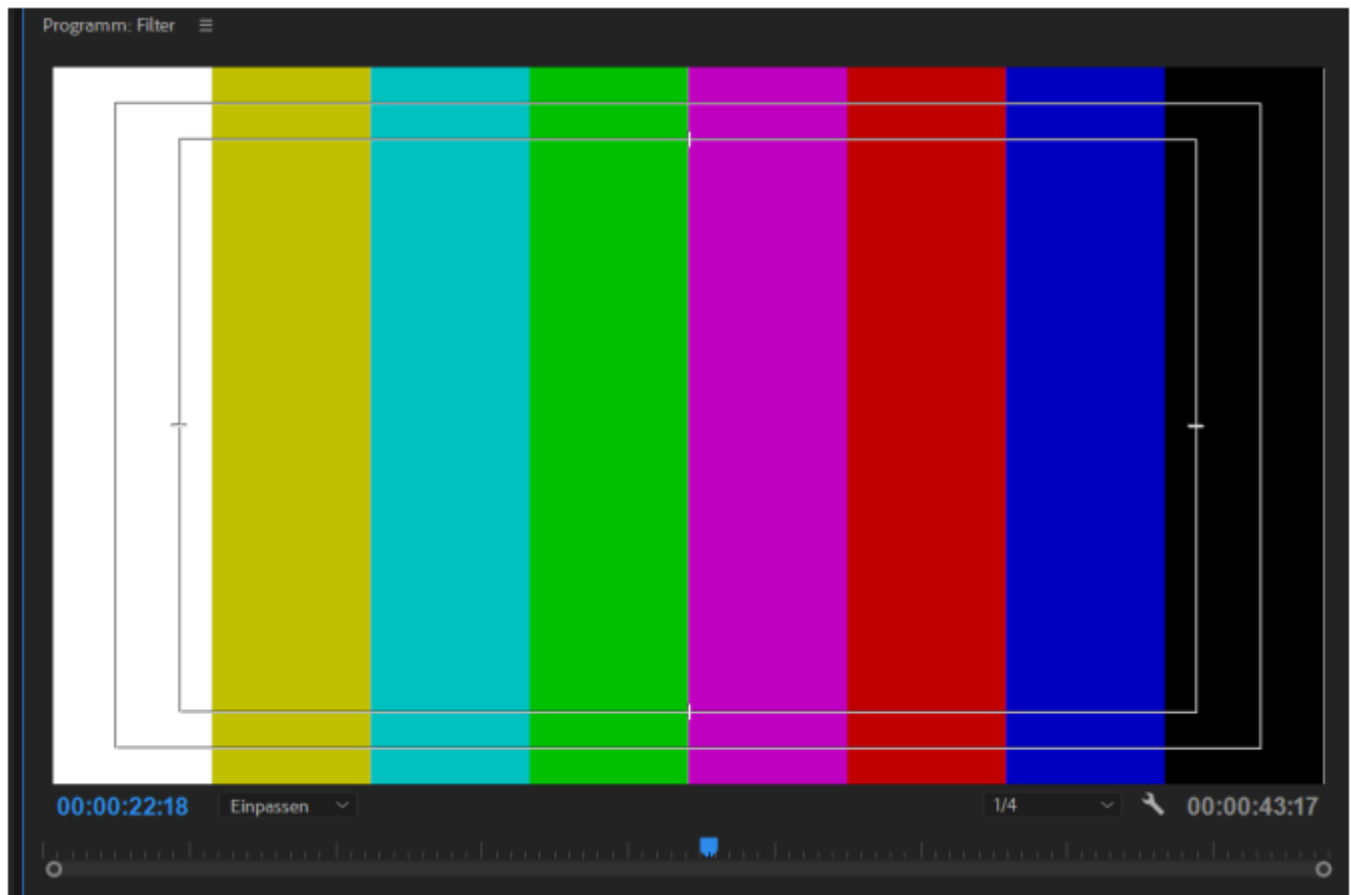


Nicht-Anamorph und Anamorph. Links ist die entzerrte Version.

Seitenverhältnis (DAR)	Beschreibung
4:3	35 mm und 16 mm Film, SD-TV
1,43:1	IMAX 70 mm Film
3:2	Fotografie
14:9	Werbung, Kompromiss zwischen 4:3 und 16:9
16:9	Video und HDTV-Standard
2,2:1	70 mm
2:1	50er Jahre Standard Amerika, RED One
2,39:1	Panavision
2,76:1	Ultra Panavision

Safe-Areas bei 16:9 Produktionen

In der Videosoftware bzw. an der Kamera können sogenannte **sichere Bereiche** eingeblendet werden. Diese sind vor allem bei der Aufnahme wichtig und definieren Pufferzonen bei der Produktion. Innerhalb dieses Bereiches wird sichergestellt, dass bestimmte wichtige Bildinformationen auch sichtbar im Bild sind. Der Bereich ist abhängig vom Seitenverhältnis.



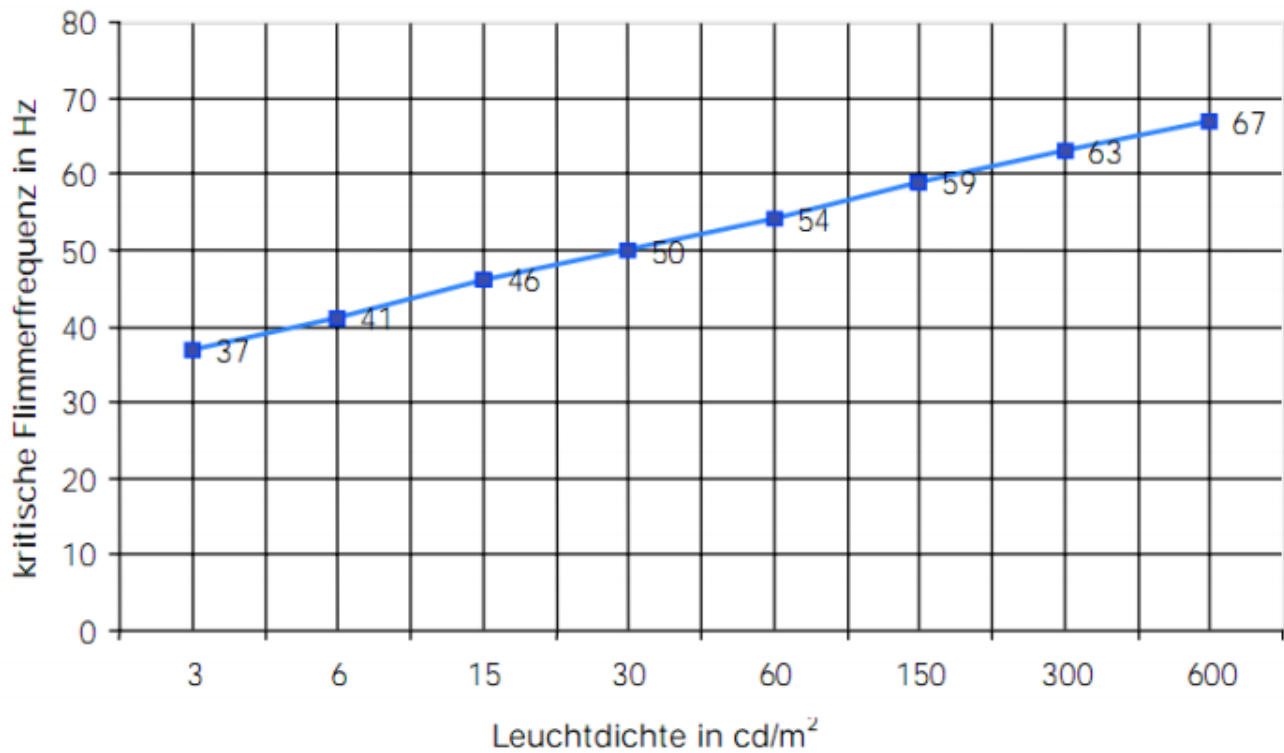
Bildverschmelzungsgrenze und Bildwiederholungsfrequenz

Die **Bildverschmelzungsfrequenz** unseres Auges (Sehsinn) liegt bei ungefähr 16 Bildern pro Sekunde (16 Hz). Für Laufbildmedien wurden folgende Werte gewählt:

- **24 Bilder/Sek** weltweit für **Film**
- **25 Bilder/Sek** in Europa; 30 Bilder/Sec in USA/Japan für **Fernsehen**

Der Unterschied zwischen Europa und USA/Japan liegt an der Verkoppelung mit der jeweiligen Frequenz des Wechselstromnetzes (Europa 50 Hz, USA/Japan 60 Hz). Obwohl in allen drei Fällen die Bildverschmelzungsfrequenz deutlich überschritten ist, ist an **hellen Bildstellen ein deutliches Flimmern** zu bemerken.

Das **Ferry-Porter Gesetz** besagt, dass die Flimmergrenzfrequenz des menschlichen Sehsinns deutlich höher als die Bildverschmelzungsfrequenz und zudem abhängig von der Helligkeit (**Leuchtdichte**) des Dargestellten ist.



Ferry-Porter Gesetz

Die **Bildverschmelzungsfrequenz** ist jene Frequenz, ab welcher eine abgebildete Bewegung als fortlaufend und zusammengehörig empfunden wird.

Die **Flimmergrenzfrequenz** ist abhängig von der Leuchtdichte und bestimmt, ob uns das Gesehene „flimmernd“ erscheint (ab 50 Hz kein Flimmern mehr bewusst bemerkbar - außer bei hellen Stellen. Siehe Ferry Porter-Gesetz).

Die **Bildwiederholfrequenz** ist die tatsächliche Anzahl an Bildern pro Zeiteinheit und kann natürlich gleich, größer oder kleiner der **Flimmerfrequenz** (= tatsächliche „Bildaufbaurate“ des Wiedergabegeräts) sein. Die Begriffe werden in der Praxis oft vermischt.

Bildwiederholrate **Empfehlungen**: • TV 50i • Film look 24p, 25PsF, evtl. 50p • Kino 24p • Sport/Action 50p

Zeilensprung und progressive Abtastung

Jedes Bild erzeugt ein Vollbild im Fall von **progressive scan**. Die Abtastung beginnt an der oberen linken Ecke des Bildes und endet an der rechten unteren Ecke.

Alternativ zum progressive scan können **bei der Aufnahme zwei Bilder zu zwei verschiedenen Zeitpunkten** abgetastet werden, wobei die Zeilen derart verschachtelt sind, dass geradzahlige und ungeradzahlige Zeilen jeweils ein eigenes sogenanntes Feld (engl. field) bilden. Dies wird als **unteres bzw. oberes Halbbild** (upper field, lower field) bezeichnet. Diese Abtastung nennt man **interlaced** (= **Zeilensprungverfahren**). Dabei entstehen ein zeitlicher Vershub und eine **höhere zeitliche Auflösung** (=flüssiger Bewegungseindruck bei gleicher Bandbreite). Dafür ist das einzelne Bild nicht so scharf und kann bei bewegungsreichen Aufnahmen kann Zeilenflimmern auftreten.



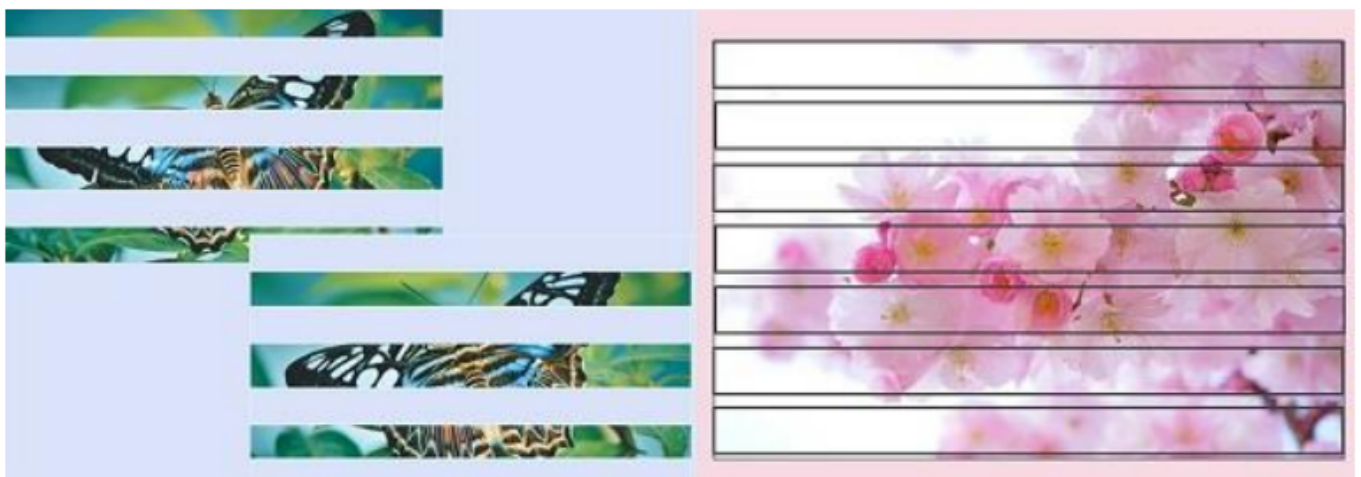
Video zu Interlaced, Progressive & Bildwiederholrat en]



Video zu Interlaced vs. Progressive]

Bei der Wiedergabe wird das interlace-Material mittels **de-interlacing** (weave, bob) in Vollbilder umgewandelt.

Achtung: Bei 50i werden 50 Halbbilder abgetastet, die Datenmenge und Bildwiederholfrequenz entsprechen technisch jedoch 25p, weshalb auch 50i-Footage eine Bildwiederholfrequenz von 25 Bildern anzeigt (siehe zB. Premiere)



Bei **PsF (Progressive Segmented Frame)** speichert man Vollbildmaterial als Halbbilder. Jedes

Vollbild wird dabei in zwei Halbbilder ohne zeitlichen Vershub der einzelnen Zeilen innerhalb eines Halbbildes geteilt.

Das HDTV-Format **1080i50** wird dann im PsF-Verfahren übertragen, wenn progressives Bildmaterial mit 25 Vollbildern pro Sekunde vorliegt und gesendet wird. Die Vorteile durch die höhere zeitliche Auflösung bei interlace-Abtastung **entfallen jedoch**.



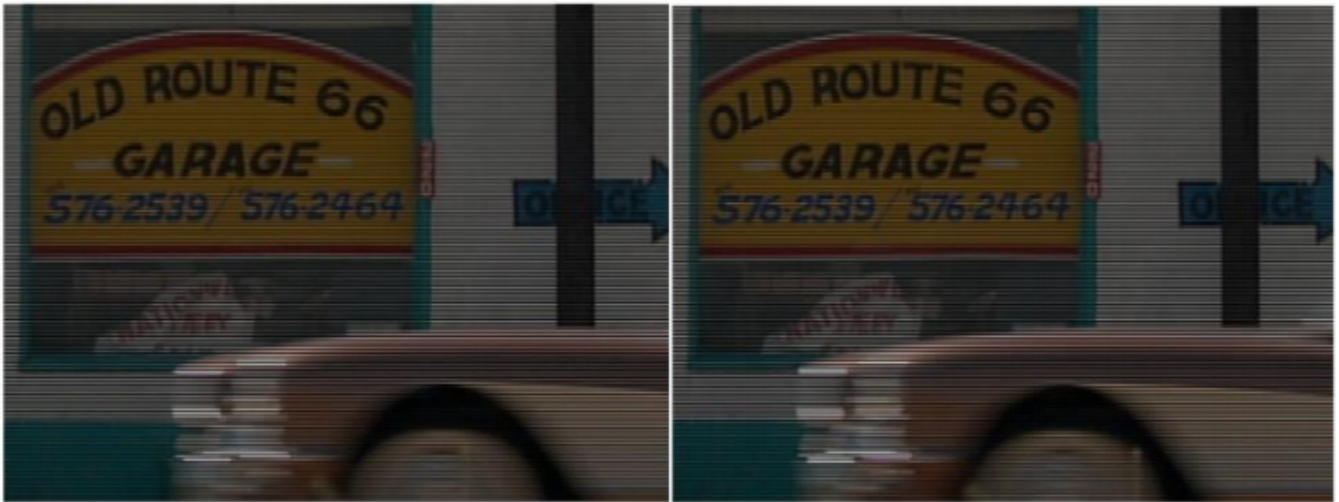
Progressive segmented Frame]

Progressive Abtastung:

- Film-Look
- Für Nachbearbeitung (Post) geeignet
- Geringe Bewegungsunschärfe
- Ortsauflösung höher (spatial resolution)

Interlace Abtastung

- Zeitliche Auflösung höher (temporal resolution)
- Video-Look
- Für Broadcast/TV-Workflow geeignet



Oberes und unteres Halbbild eines Frames



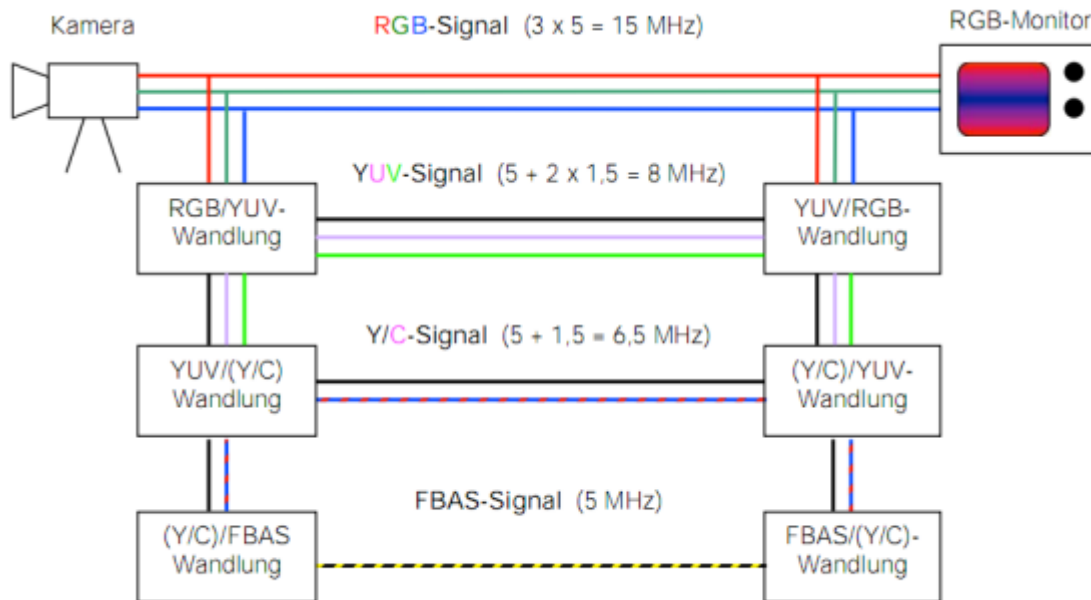
Links: Zwei Halbbilder vereint (50i), Rechts: Progressive Abtastung (25p)

[Weitere Infos](#)

Analoge Videosignale

Trotz voranschreitender Digitalisierung finden wir im videotechnischen Bereich noch viele analoge Signale und Technologien, welche neben den digitalen Übertragungen in den folgenden Kapiteln dargestellt werden.

Primär ist das **RGB-Komponentensignal** von Bedeutung. In der folgenden Abbildung wird der Signalfuss von RGB bis FBAS dargestellt.



RGB-Signal

Diese Signalform ist meist der **Ausgangspunkt für die Übertragung von Videosignalen**. In der Theorie der analogen Videotechnik ist RGB die Video-Übertragungsnorm, die die höchste Bildqualität liefert, d.h. die geringsten Übertragungsverluste erzeugt. Doch in der Heimkino-Praxis liegt das Problem von RGB in seiner Variantenvielfalt. Bei dem im Computerbereich üblichen RGB HV werden zusätzlich noch die horizontalen (H) und vertikalen (V) Sync-Signale auf je einer weiteren Leitung getrennt übertragen. Im Videobereich wird das Sync-Signal mitunter auf nur einer separaten Leitung zusammengefasst oder auf Grün aufmoduliert (Sync-on-Green) - dann bleibt es bei drei Leitungen. Neueste Komponenten kommen ganz ohne Sync aus. Rot, Grün und Blau nehmen einen **Pegelbereich von 0...700 mV** ein; die Bandbreite beträgt jeweils 5 MHz.



YUV-Komponentensignal

YUV, auch Komponenten- (vom englischen „Components“) Video genannt, ist das amerikanische Pendant zu RGB und funktioniert auch ähnlich. Wie RGB können Sie auf den ersten Blick auch YUV an drei rot/grün/blauen Video-Cinch-Buchsen erkennen. Dennoch ist das Signal nicht kompatibel zu RGB. Beim YUV-Signal wird die Bildinformation in ein Helligkeitssignal (Y = grün markierte Buchse) und zwei Farbdifferenzsignale (Cb = blau markierte Buchse und Cr = rot markierte Buchse) zerlegt.

Die **Farbdifferenzsignale werden wie folgt gebildet: $U = R - Y$ und $V = B - Y$** . Aus $G - Y$ ergäben sich nur geringe Pegel, da der Grünauszug dem Y-Signal sehr ähnlich ist. Aus diesem Grund genügt es, nur zwei Farbdifferenzsignale aus $R - Y$ und $B - Y$ zu bilden, um die gesamten Farbinformationen zu gewinnen.

YPBPR-Komponentensignal

Dieses Signal ist ähnlich dem YUV-Signal, nutzt jedoch einen **anderen Farbraum** und wird bei **Component-Video**-Anschluss übertragen bzw. als Ausgangspunkt für digitale Signale genutzt.



Y/C-Signal (S-VIDEO)

Y/C (auch bekannt als Separate Video, S-Video) bezeichnet die getrennte Übertragung von Luminanz (=Helligkeit)- und Chrominanz(Farbinformation)-Informationen. Die beiden **Farbdifferenzsignale U und V werden zu einem Chrominanzsignal (C)** gemischt. Luminanz- und Chrominanzsignal werden getrennt

übertragen (Y/C). Es ermöglicht, vor allem beim Einsatz in preisgünstigen Geräten, qualitativ bessere Signale als Composite Video (FBAS), erreicht jedoch nicht die Qualität von RGB-Signalen oder Komponentensignalen.



Dieser Stecker ist ein **4-poliger DIN-Stecker**



FBAS-Signal

Composite-Video - oft auch FBAS oder einfach nur „Video“ genannt- ist die einfachste der Videoverbindungen. Erkennbar ist FBAS an der meist gelb markierten Cinch-Buchse am Gerät. Bei FBAS wird das gesamte Videosignal mit allen seinen Komponenten wie z.B. Farb-, Helligkeits- oder Synchron-Signalen auf eine einzige Leitung moduliert. Daher stammt auch der zweite Name für diese Signalvariante, nämlich „Composite“, was soviel heißt wie „alles in einem zusammengefasst“. Entsprechend schlecht und störanfällig ist FBAS im Vergleich zu allen anderen Signalarten.



Cinch-Stecker gelb für FBAS, rot und weiß für Audio



FBAS Stecker und Buchse für Antennensignal

Gammakorrektur für RGB-Signale

Aus den RGB-Signalen vom Sensor wird mittels Gammakorrektur ein R'G'B'-Signal erzeugt. Die **Gammakorrektur** ist nötig, damit die linear anwachsenden Größen die der Sensor liefert in ein dem menschlichen Sehsinn entsprechendes, nicht-lineares Signal konvertiert wird.

Die menschliche Wahrnehmung ist anatomisch so, dass dunklere Tonwerte heller und helle Tonwerte weniger hell wahrgenommen werden. Die Empfindlichkeit von Bildwandlern ist jedoch in den Lichtern höher als in den Tiefen, weshalb nach der Gammakorrektur mehr Tonwerte für die Tiefen zur Verfügung stehen.

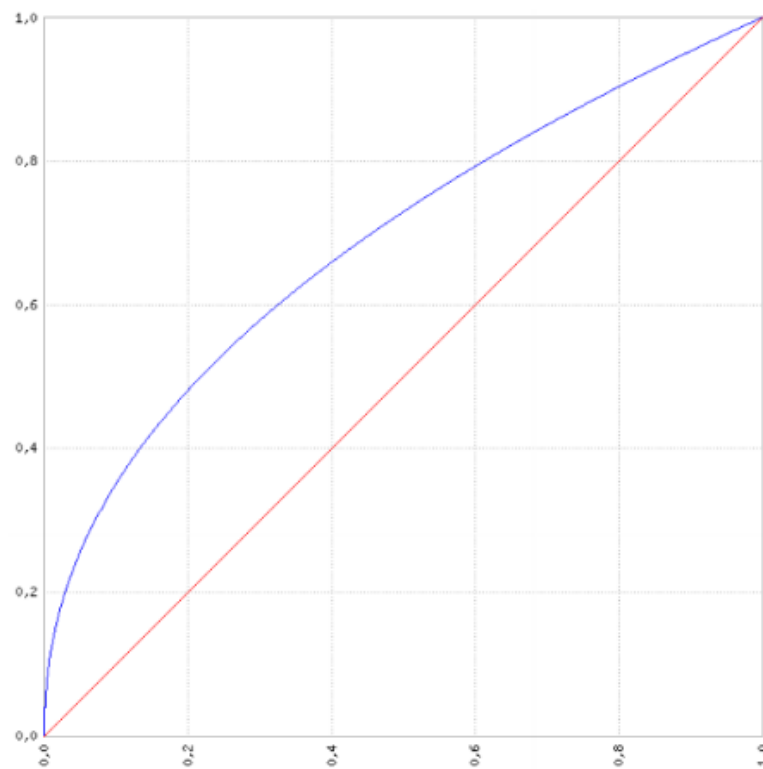
Originalszene



Linear kodiert



Gammakorrigiert



Gammakurve (blau) für ein Gamma von 2,2; Lineare Werte des Bildwandlers (rot)



Links linear und rechts mit Gamma 2,2

Digitale Videosignale

Das analoge Y'P'bP'r-Signal wird mit der entsprechenden Abtastfrequenz digitalisiert und serialisiert.

Komprimierung

Bei der digitalen Kodierung wird meist eine Komprimierung zur **Reduktion der Datenmenge** angewandt. Dabei unterscheidet man zwischen **verlustfreier** (lossless) und **verlustbehafteter** (lossy) Komprimierung.

Digitale Komponentensignal YCBCR

Das YCbCr-Farbmodell entspricht weitestgehend dem in der Analogtechnik eingesetzten YUV-Farbmodell, resp. dem YPbPr-Farbmodell. Es wird in JPEG und der MPEG-Kompression benutzt, die in DVDs, Digital-TV, Video-CDs und in Camcordern eingesetzt werden und kann über die HDMI-Schnittstelle und die DVI-Schnittstelle übertragen werden. YCbCr ist ebenso wie das RGB-Farbmodell im ITU-Standard BT.601 beschrieben.

SDI - Serial Digital Interface

SDI wird von der SMPTE spezifiziert und ist eine Weiterentwicklung der analogen PAL- und

NTSC-Standards. Über alle **HD-SDI-Verbindungen** können **Bild, Ton und Metadaten** übertragen werden.

Es dient primär zur **Übertragung** von **unkomprimierten** und **unverschlüsselten Videodaten** über **Koaxialkabel** oder **Lichtwellenleiter**. Es kommt hauptsächlich im Bereich **professioneller Fernsehstudios** und im Bereich von **Fernsehsendern** zum Einsatz.

Der Stecker für dieses Kabel heißt BNC und verfügt über einen **Bajonettverschluss**, der verhindert, dass sich Stecker und Buchse versehentlich lösen.



HDMI

HDMI ist in verschiedenen Spezifikationen im Laufe der Zeit erweitert worden und eine semi-professionelle AV (Audio & Video)-Schnittstelle. Folgende Übersicht zeigt die wichtigsten Features.

HDMI-Version	1.0	1.1	1.2	1.3	1.4	2.0	2.0a	2.1
sRGB	✓	✓	✓	✓	✓	✓	✓	✓
YCbCr	✓	✓	✓	✓	✓	✓	✓	✓
8-Kanal-PCM, 192 kHz, 24-bit	✓	✓	✓	✓	✓	✓	✓	✓
CEC ⁸ (Steuerfunktionen wie Play, Standby...)	✓	✓	✓	✓	✓	✓	✓	✓
Deep Color (30-bit-Farbtiefe)	✗	✗	✗	✓	✓	✓	✓	✓
Auto Audio to video synchronization lip-sync	✗	✗	✗	✓	✓	✓	✓	✓
Dolby-TrueHD-Bitstream-Fähigkeit	✗	✗	✗	✓	✓	✓	✓	✓
DTS-HD-Master-Audio-Bitstream-Fähigkeit	✗	✗	✗	✓	✓	✓	✓	✓
3D	✗	✗	✗	✓	✓	✓	✓	✓
1920 × 1080 (1080p), 120 Hz	✗	✗	✗	✗	✓	✓	✓	✓
Ethernet channel	✗	✗	✗	✗	✓	✓	✓	✓
Audio return channel (ARC)	✗	✗	✗	✗	✓	✓	✓	✓
4K	✗	✗	✗	✗	✓	✓	✓	✓
4K 50/60p	✗	✗	✗	✗	✗	✓	✓	✓
4K 3D	✗	✗	✗	✗	✗	✓	✓	✓
32-Kanal-Audio-Support	✗	✗	✗	✗	✗	✓	✓	✓
1536-kHz-Audio-Support	✗	✗	✗	✗	✗	✓	✓	✓
1080p 3D HFR	✗	✗	✗	✗	✗	✓	✓	✓
21:9 Cinemascope support	✗	✗	✗	✗	✗	✓	✓	✓
Multi Stream Audio und Video	✗	✗	✗	✗	✗	✓	✓	✓
High Dynamic Range	✗	✗	✗	✗	✗	✗	✓	✓
4K 3D 50/60p support	✗	✗	✗	✗	✗	✗	✗	✓
Unterstützung von 14 und 16 Bit Farbtiefe	✗	✗	✗	✗	✗	✗	✗	✓



HDMI Typ A, Typ C (Mini), Typ D (Micro)

Grundwissen Film/Video

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:2

Last update: **2019/10/01 09:16**



3) Webtechniken - HTML & CSS

- [3.1\) Schichtenmodell einer Webseite](#)
- [3.2\) DOM](#)
- [3.3\) HTML](#)
- [3.4\) CSS](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3

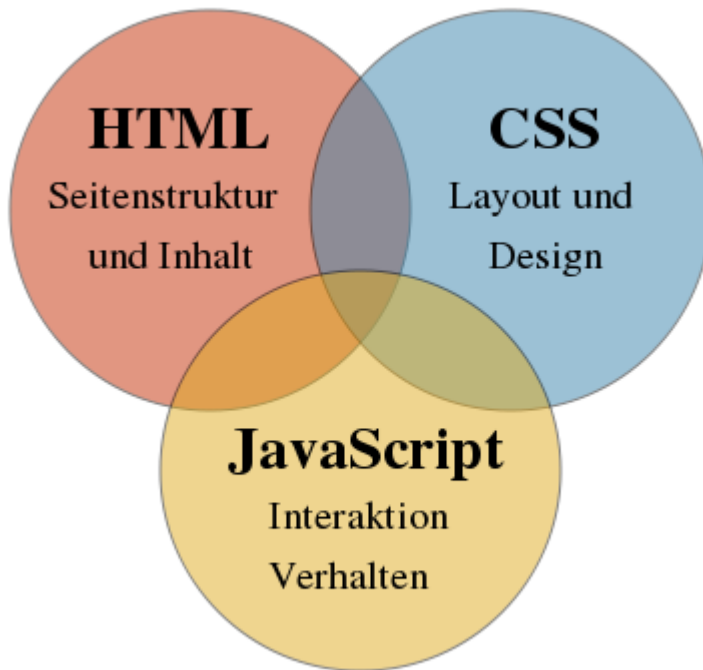


Last update: **2019/10/15 08:11**

HTML und CSS und Scripts

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**)



Dies entspricht auch dem in der Informatik und anderen Wissenschaftsbereichen anerkannten **Prinzip der Trennung von Zuständigkeiten (Separation of Concerns)**, die besagt, dass verschiedene Elemente der Aufgabe möglichst in verschiedenen Elementen der Lösung repräsentiert werden sollten. Um Webseiten möglichst effizient und einfach zu entwickeln sowie sie nachträglich mit geringem Aufwand pflegen zu können, sollten diese Aufgaben strikt voneinander getrennt werden:

- Im **Markup** werden keine Angaben zur Präsentation oder Interaktion gemacht.
- Im **Stylesheet** befinden sich demnach alle Angaben zur Präsentation in möglichst effizienter Weise.
- **JavaScript** wird nicht mit HTML-Attributen aufgerufen, sondern dynamisch an HTML-Elemente angehängt.

HTML - Struktur für den Inhalt

Der Inhalt von Webseiten besteht grundsätzlich aus einem Text-Dokument. HTML soll diese Texte sinn- und bedeutungsvoll strukturieren, indem z.B. Überschriften, Listen, Absätze, Datentabellen, zusammenhängende Bereiche sowie wichtige Abschnitte, Zitate usw. als solche ausgezeichnet werden.

Empfehlung

Verwenden Sie möglichst bedeutungsvolle HTML-Elemente, die sich am Inhalt der Webseite orientieren. Vergeben Sie für die Feinstrukturierung sparsam gesetzte und aussagekräftige Klassen und IDs. Der Klassenbezeichner hinweis etwa ist semantischer als rot-unterstrichen. Klassen sollten den Grund ihrer Existenz in ihrem Namen tragen, nicht ihr visuelles Erscheinungsbild. Vermeiden Sie zusätzliche Elemente wie `<div class=„container“>`, die allein für die Gestaltung verwendet werden. Eine Webseite mit weniger Elementen verbessert die Ladezeit und damit die Benutzerfreundlichkeit.

CSS - Formatierung und Gestaltung

CSS ist dafür da, die Regeln für Darstellung dieser Inhalte vorzugeben, sei es auf einem Smartphone, einem Desktop-Bildschirm, ausgedruckt auf Papier oder anders. Optimalerweise wird es in einem ausgelagerten Stylesheet notiert, das dann alle Seiten Ihres Webauftritts zentral formatiert. Alternativ können Sie so das Design Ihrer Webseite schnell und unkompliziert ändern. Das klassische Beispiel dafür ist der CSS ZEN GARDEN, in dem eine Webseite mit verschiedenen Stylesheets auf Knopfdruck immer wieder ein völlig anderes Layout bekommt.

Empfehlung

Setzen Sie Angriffspunkte für CSS-Selektoren wie Klassen und IDs nur sparsam ein - alternativ können Sie Elemente mit strukturelle Pseudoklassen wie `:first-child` etc. ansprechen, allerdings sind strukturelle Pseudoklassen bei Markup-Änderungen weniger robust als andere Selektoren.

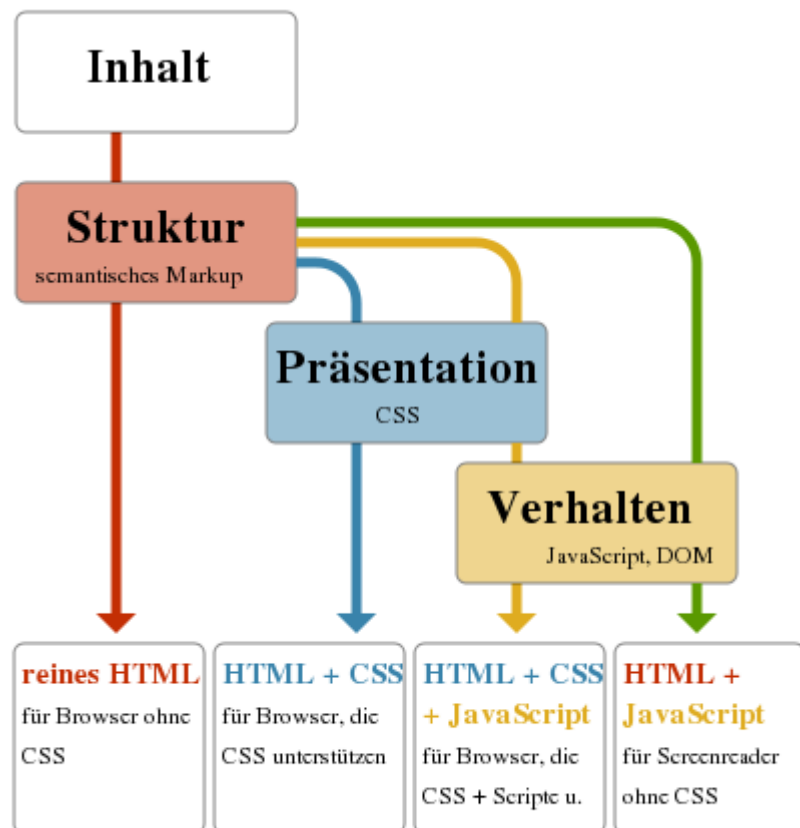
JavaScript - Interaktivität/Verhalten

Wenn nun die dritte Technik – JavaScript – hinzutritt, sollten Sie dieses Modell konsequent fortführen. JavaScript kommt in diesem Konzept die Aufgabe zu, dem Dokument Verhalten (behaviour) hinzuzufügen. Damit ist gemeint, dass das Dokument auf gewisse Anwenderereignisse reagiert und z.B. Änderungen im Dokument vornimmt (DOM-Scripting).

Das Schichtenmodell

Da diese drei Aspekte aufeinander aufbauen (mit der Struktur als Basis), kann man sie auch als Schichtenniodell betrachten (siehe Abbildung). Hier wird deutlich, dass die HTML-Struktur die Basis bildet, auf der die darüberliegenden Schichten aufsetzen.

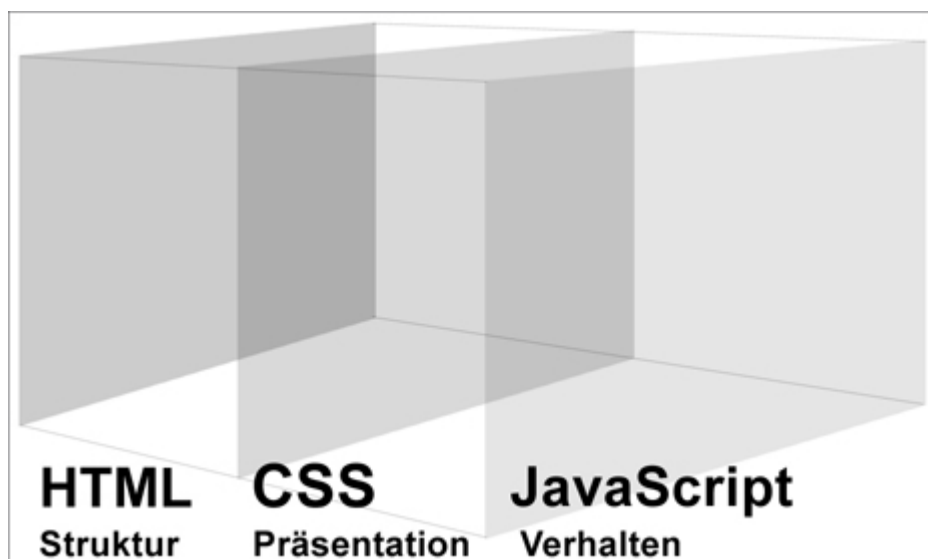
Tatsächlich liest der Browser auch zunächst diese Struktur ein, die er anschließend gemäß den CSS-Anweisungen »dekoriert;« und schließlich mit dem in Javascript beschriebenen Verhalten ausstattet. Setzen wir uns kurz mit den Grundlagen von HTML auseinander.



Die wichtigste Schicht eines Webdokuments ist das Markup. Dessen Inhalte müssen auch ohne Stylesheets oder JavaScripte sicht- und benutzbar sein.

Durch Stylesheets kann eine weitere Schicht über das Dokument gelegt werden, die ein Layout zuschaltet, das die Lesbarkeit verbessert und durch die Gestaltung die Webseite ansprechender und interessanter macht. Falls das Stylesheet nicht geladen werden kann, bleibt die Webseite aber weiterhin benutzbar.

JavaScript bildet die dritte Schicht, die dem Dokument Interaktivität und Verhalten gibt. Bei abgeschaltetem oder nicht verfügbaren JavaScript (z.B. bei Screenreadern) bleibt die Seite aber weiterhin benutzbar.



Die drei Ebenen - Struktur, Präsentation und Verhalten

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_01



Last update: **2019/10/15 08:18**

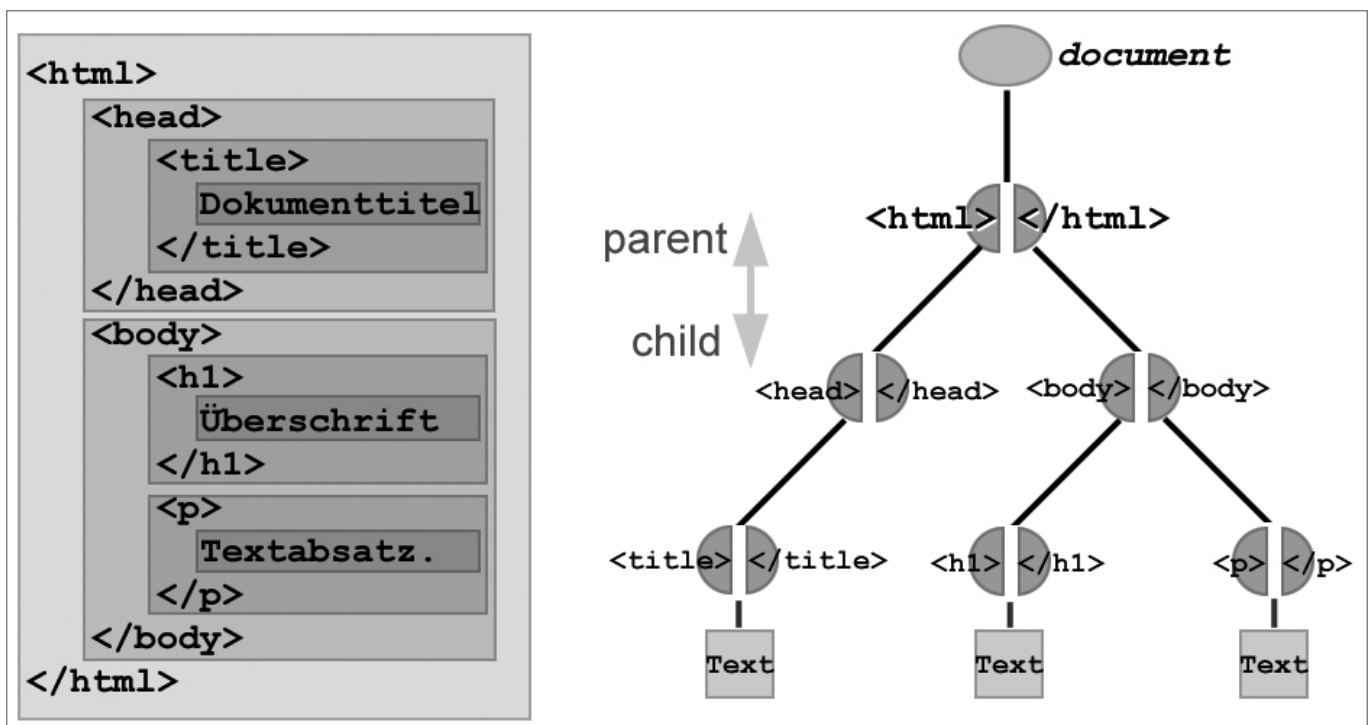
Das Document Object Model

Ein wichtiges Konzept im Zusammenhang mit dem Zugriff auf Webseiten ist das Document Object Model (.DOM). Das DOM stellt die Schnittstelle zwischen der Dokumentstruktur und dem Rest dar und dient Ihnen einerseits zur Verknüpfung von HTML und CSS, andererseits ist es auch das Interface, das Ihnen erlaubt, per Script auf Struktur und Präsentation des Dokuments einzuwirken.

Das Erstellen des DOM-Baums

Das DOM stellt eine Abstraktion des Dokuments dar, die der Browser beim Einlesen des HTML-Quelltexts im Arbeitsspeicher erzeugt. Eigentlich existiert es also rein virtuell. Stellen Sie es sich als Baumstruktur vor, und zwar in Form eines nach unten hängenden Baums, der an einem Punkt aufgehängt ist und sich ab dort verzweigt. Bezeichnen wir den Punkt, an dem der Baum ansetzt, als Dokumentknoten (documentNode). Dieser Dokumentknoten besitzt keine Entsprechung im Dokument, sondern stellt das »Dokument« an sich dar.

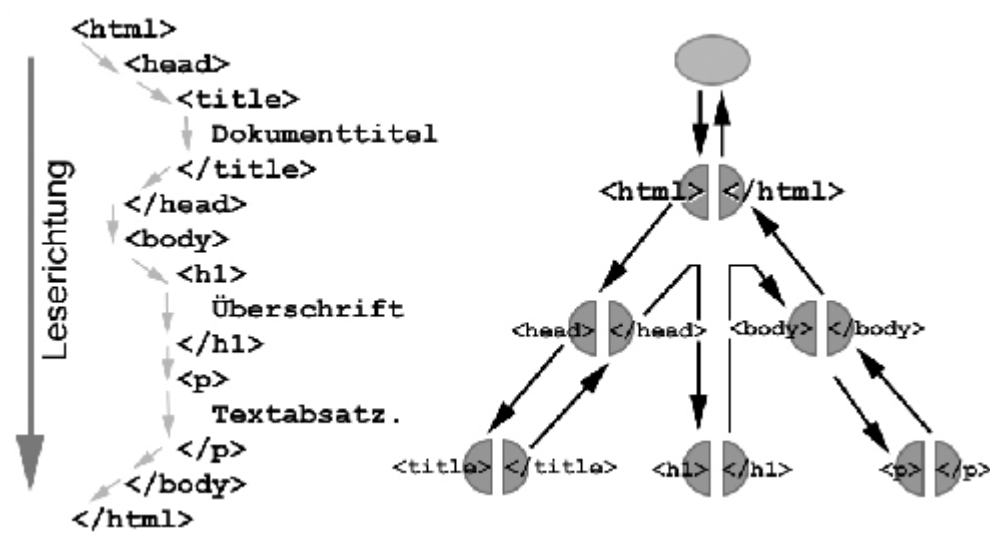
Einen solchen Dokumentknoten erzeugt der Browser beim Einlesen (Parsing) des Dokuments. Er hängt dann für jedes Element, das er in Quelltextreihenfolge antrifft, einen weiteren Knoten unten an diesen Dokumentknoten an. Das erste Element, das er antrifft, ist stets das Root-Element HTML (allgemein als document-Element bezeichnet). Ab hier spaltet sich der Baum in zwei Zweige, die den <head> und den <body> repräsentieren (siehe Abbildung).



Der HTML-Quelltext und seine DOM-Repräsentanz

Der Baum besteht also aus Verbindungen und Knoten. Im Falle eines Elementknotens fallen in diesem Start- und Endmarke des Quelltexts zusammen. Textknoten bilden stets das Ende eines Zweiges (»leaf nodes«). Enthält ein Element einen Inhalt (gehen wir also hierarchisch in dessen Inneres), fächert sich der Baum weiter nach unten auf. Hierdurch entsteht für jeden Bestandteil dieses Inhalts ein weiterer Ast mit daran hängenden Knoten.

Hierbei gilt der oben liegende Knoten als »Elternknoten« (»parent«), die von ihm unmittelbar abstammenden Knoten als »Kindknoten« (»children«). Sowohl Elemente als auch Textknoten stehen stets in einer Eltern-Kind-Beziehung, wobei jeder Knoten genau einen Elternknoten besitzt (niemals mehrere). Ein Element-knoten besitzt darüber hinaus weitere Eigenschaften, zu denen (salopp gesprochen) auch seine Attribute gehören. Der Browser baut auf diese Weise sukzessive ein komplettes Abbild der hierarchischen Struktur des Dokuments auf.



Das DOM ist also eine Abstraktion folgender Information:

- Wie ist der hierarchische Zusammenhang in der Dokumentstruktur?
- Welches Element ist an welcher Position der Hierarchie?
- Welche Eigenschaften hat es (z. B. Attribute, Inhalte, Nachbarelemente)?

Hier soll kurz ein Vorteil des DOM-Konzepts gegenüber »seriellem« Quelltext erwähnt werden, der darauf beruht, dass ein Knoten gleichzeitig Start- und Endmarke eines HTML-Tags repräsentiert: Bei einer Manipulation des Baums (wir werden gleich sehen wie das geht), also der Entnahme oder dem Hinzufügen einzelner Knoten oder ganzer Zweige, wird stets mit vollständigen Strukturen gearbeitet. Ein Dokument behält so stets seine Wohltgeformtheit, also die Art von regelmäßiger Struktur, die in der XMT.-Datenverarbeitung gefordert ist.

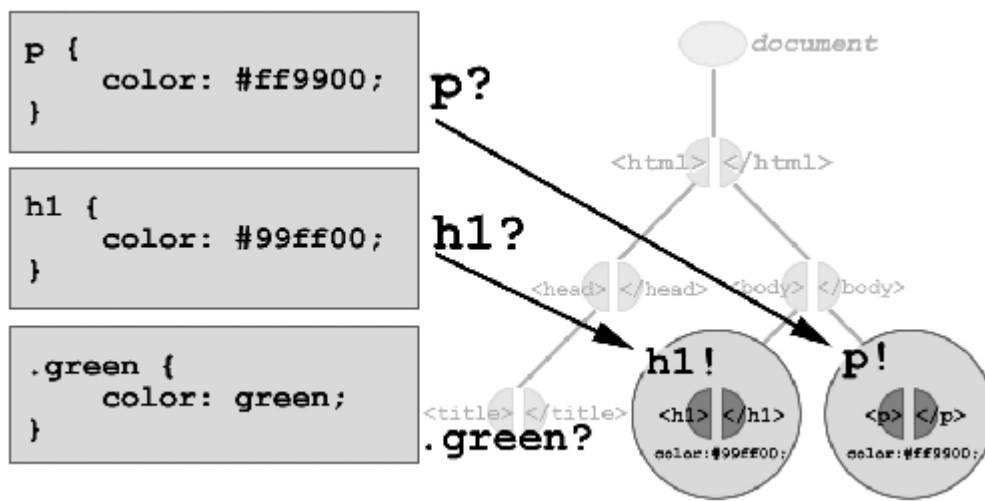
Das »Schmücken« des DOM-Baums

Dieses Modell, in dem alle Elemente des Dokuments repräsentiert sind, ist noch eine Abbildung der »nackten« Struktur - durchaus wörtlich zu nehmen, denn bis jetzt hat der Browser die CSS-Informationen noch nicht eingebracht. Die nun folgende Phase wird als »decorating the tree« bezeichnet: Der Browser liest alle CSS-Informationen ein und löst dabei auftretende Konflikte und Unstimmigkeiten auf. Anschließend liegt das sogenannte Stylesheet vor: die vollständigen Präsentationsvorschriften, die auf das Dokument angewendet werden sollen.

Nun werden die Selektoren aller CSS-Regeln registriert und der Dokumentbaum anhand dieser Vergleichsmuster durchsucht. Jeder Elementknoten, auf den das Muster zutrifft, bekommt den Rcglsatz zugewiesen. Auf diesem Weg werden alle Regeln durchgearbeitet. Regeln, für die keine Übereinstimmung mit dem Dokument gefunden wird, liegen brach (siehe folgende Abbildung).

Bekommt ein Element hierbei mehrfach Regeln zugewiesen, addieren sich diese. Ein dabei

auftretender Konflikt wird nach Rang der Regel beigelegt. Sobald der Vorgang beendet ist, ist der Baum »dekorierte und der Browser geht daran, ihn im Viewport darzustellen.



Der DOM-Baum wird per Vergleichsmuster durchsucht

Im dekorierten Baum sind folgende Informationen zusätzlich in den einzelnen Elementknoten gespeichert:

- Welche Präsentationsvorschriften existieren für dieses Element?

Die Stilevorschriften können wir uns durchaus als »Eigenschaften« des Elements vorstellen, die beim Elementknoten gelagert werden. Wir werden gleich sehen, dass dies uns hilft, die CSS-Regeln eines Elements auszulesen und auch bei Bedarf zu ändern. In diesem Augenblick kommt wieder JavaScript ins Spiel.

Manipulation von DOM und CSS per JavaScript

Der dekorierte Baum stellt, wie wir jetzt wissen, ein Abbild des Dokuments im Arbeitsspeicher des Browsers dar, das der Darstellung des Dokuments im Viewport entspricht. Da es sich um eine rein virtuelle Sache handelt, ist dieses Abbild per Programmierung beliebig manipulierbar. Genau dies ist überhaupt die Aufgabe des DOM- eine Schnittstelle (API) zu bieten, die es ermöglicht, mittels einer Programmiersprache auf das Dokument einzuwirken.

In Zusammenhang mit JavaScript bietet das DOM eine Reihe von Schnittstellenfunktionen, die eine Brücke schlagen zwischen der Scripting-Umgebung und dem Dokumentbaum. Einige dieser Methoden sind dem Dokumentknoten (den wir hierfür praktisch erweise einfach als JavaScript-Objekt betrachten! unterstellt. Andere Methoden stehen auch direkt den Elementknoten zur Verfügung.

Diese DOM-Methoden sind nicht ausgesprochen zahlreich und zum Teil umständlich anzuwenden - ihre Anzahl hängt zudem von der Implementierung des DOM ab (»DOM-Level«), die der ausführende Browser jeweils unterstützt.

Dies sind ihre Aufgaben:

- Selektieren von Knoten
- Traversieren (Bewegung) innerhalb des Baums
- Wert (Inhalt) eines Knotens auslesen oder schreiben

- Attribute eines Elementknotens lesen oder schreiben
- Erzeugen von Element- und Attributknoten
- Einhängen von Knoten in den Baum
- Löschen von Knoten im Baum

DOM-Methode	Erläuterung
<code>addEventListener()</code>	Bindet Event Handler an DOM-Element.
<code>removeEventListener()</code>	Löst Event Handler an DOM-Element.
<code>createAttribute()</code>	Erzeugt einen Attributknoten.
<code>createTextNode()</code>	Erzeugt einen Textknoten.
<code>createElement()</code>	Erzeugt einen Elementknoten.
<code>getAttribute()</code>	Liest einen Attributwert.
<code>setAttribute()</code>	Schreibt einen Attributwert.
<code>removeAttribute()</code>	Entfernt einen Attributknoten.
<code>appendChild()</code>	Hängt Elementknoten am Ende des Inhalts des aktuellen Knotens ein.
<code>insertBefore()</code>	Hängt Elementknoten vor dem aktuellen Knoten ein.
<code>removeChild()</code>	Entfernt Kindknoten des aktuellen Knotens.
<code>replaceChild()</code>	Ersetzt Kindknoten des aktuellen Knotens.
<code>getElementById()</code>	Referenziert einen Elementknoten per ID.
<code>getElementsByName()</code>	Erstellt NodeList aus Elementknoten nach CSS-Klasse.
<code>getElementsByTagName()</code>	Erstellt NodeList aus Elementknoten nach Tag-Bezeichner.

Die wichtigsten DOM-Methoden zur DOM-Manipulation

Links:

- [Was ist DOM ?](#)
- [Der DOM-Baum](#)
- Eine umfassende Onlinereferenz zu DOM (in englischer Sprache) findet man bei der Mozilla Foundation unter dieser Adresse <https://developer.mozilla.org/en/DOM>

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_02

Last update: **2019/10/15 08:09**



Grundlagen in HTML

- [Einführung](#)
- [Grundgerüst eines HTML-Dokuments](#)
- [Formatieren von Texten](#)
- [Listen](#)
- [Tabellen](#)
- [Hyperlinks](#)
- [Einbinden von Grafiken](#)

-
- [Aufgaben](#)

-
- [Grundlagen JavaScript](#)
 - [Beispiele JavaScript](#)

Links:

- [SELFHTML, die HTML-Referenz](#)
- [Die Zukunft: HTML5](#)
- [W3-Schools](#) (engl.)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_03

Last update: **2019/10/15 08:10**



3.4) CSS

- [3.4.1\) Grundlagen](#)
- [3.4.2\) Deklarationen](#)
- [3.4.3\) Vererbung und Kaskadierung](#)
- [3.4.4\) XHTML-Elemente gestalten](#)
- [3.4.5\) Übungen](#)
- [3.4.6\) Links](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04



Last update: **2019/11/17 16:02**

3.4.1) Grundlagen

- [3.4.1.1\) Was ist CSS?, Geschichte und Vorteile von CSS](#)
- [3.4.1.2\) Einbindung in \(X\)HTML](#)
- [3.4.1.3\) Probleme](#)
- [3.4.1.4\) Einheiten](#)
- [3.4.1.5\) Block- und Inlineelemente](#)
- [3.4.1.6\) Exkurs: Dokumenttyp-Definitionen](#)
- [3.4.1.7\) Exkurs: Unterschiede zwischen HTML und XHTML](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_01



Last update: **2019/11/17 15:43**

3.4.1.1) Cascading Styleseets



Was ist CSS?

CSS steht für Cascading Stylesheets, „kaskadierende Stilunterlagen“. CSS wird zum Designen eines (X)HTML-Layouts (Struktur) verwendet. CSS ist somit eine Möglichkeit, das Aussehen einer Website zu steuern, und bietet dazu unendlich an Attributen für die (X)HTML-Elemente.

Geschichtlich

1994 wurde der erste Entwurf von CSS veröffentlicht, 1996 wurde es schließlich vom W3C als Empfehlung freigegeben. (W3C: Word-Wide-Web-Consortium, es veröffentlicht Empfehlungen die als „De-facto-Standards“ fungieren. Tatsächlich Standards vorgeschrieben werden vom W3C jedoch nicht.)

Die neue Version CSS3 ist aktuell in Entwicklung, und wird viele neue Möglichkeiten zur Gestaltung von Websites bieten. Die Spezifikation wird als *working draft* eingestuft. Viele Browser unterstützen in aktuellen Versionen aber schon viele Funktionen von CSS3.

Vorteile von CSS

- Trennung von Inhalt und Layout möglich
- Einfachere Wartung, Änderung in CSS-Datei gilt sofort für gesamte Website mit allen Unterseiten
- Suchmaschinenoptimierung einfacher, da die Crawler den CSS-Code nicht lesen und sich somit ganz auf den Inhalt der Website konzentrieren können
- Erleichterte Barrierefreiheit
- Deutlich mehr Möglichkeiten zur Formatierung als mit (X)HTML.

Beispiele

Beispiele für die Verwendung von CSS sind auf der Website <http://www.csszengarden.com> zu finden.

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_01:3_04_01_01



Last update: **2019/11/17 15:57**

3.4.1.2) Einbindung in (X)HTML

CSS-Eigenschaften können im wesentlichen auf 3 Arten mit (X)HTML verwendet werden:

Externe Styles: Einbindung einer CSS-Datei im <head>-Bereich

Die CSS-Eigenschaften werden in eine externe Datei, meist mit der Endung „.css“ ausgelagert. Diese Datei wird dann im <head>-Bereich der (X)HTML-Seite mit einem <link>-Tag eingebunden.

```
<head>
<-- andere Head-Angaben -->
<link rel="stylesheet" type="text/css" href="style.css" />
</head>
```

Interne Styles: Definieren der CSS-Eigenschaften im <head>-Bereich

Bei dieser Methode werden die CSS-Eigenschaften nicht ausgelagert, sondern direkt im <head>-Bereich festgelegt. Dies geschieht innerhalb eines style-Tags.

```
<head>
<-- andere Head-Angaben -->
<style type="text/css">
/* <![CDATA[ */
h2 {
    background-color: #E6DACF;
    color: #806040;
}
p {
    background-color: #CFDCE6;
    color: #406480;
}
/* ]]> */
</style>
</head>
```

Inline-Styles: CSS-Eigenschaften direkt im (X)HTML-Tag festlegen

Diese Methode eignet sich besonders, wenn es nur 1-2 CSS-Formatierung für die ganze Seite gibt. Dann wird die Eigenschaft direkt im Tag innerhalb eines style-Attributs festgelegt.

`Das ist ein Text.`

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_01:3_04_01_02



Last update: **2019/11/17 15:57**

3.4.1.3) Probleme

Unterschiedliche Implementierungen in Browsern

Die CSS-Codes werden vom Browser verarbeitet, somit hängt die Darstellung von der jeweiligen Implementierung ab. Häufig sind manche CSS-Eigenschaften in manchen Browsern nicht implementiert, obwohl sie schon lange Standard sind - andererseits werden aber auch Eigenschaften implementiert, die noch nicht standardisiert wurden. Auch kommt es immer wieder vor, dass die Browser die CSS-Codes unterschiedlich interpretieren, also sieht eine Website im schlimmsten Fall in jedem Browser anders aus. Ähnlich gestaltet sich dieses Problem auch bei Javascript und manchen HTML-Tags. Abhilfe gibt es mit Browser-Weichen (kommt später) und CSS-Hacks (kommt nicht).

Größere Probleme in dieser Hinsicht gab es in den letzten Jahren beim Internet Explorer, weil Microsoft häufig eigene Implementierungen verwendete. Seit einiger Zeit beteiligt sich Microsoft selbst am W3C und will ebenfalls deren Empfehlungen umsetzen. Mit dem in naher Zukunft erscheinenden Internet Explorer 9 gelingt das weitgehend, während es im IE8 und IE7 schon Fortschritte gab.

Browserspezifische Eigenschaften

Manchmal bauen Browserhersteller für ihren Browser spezifische Eigenschaften ein, ein Beispiel wäre die Eigenschaft **-moz-boarder-radius:**, mit dieser ist es möglich, dass Boxen mit abgerundeten Ecken dargestellt werden. Auch der Browser Safari hat hierzu eine eigene Eigenschaft, **-webkit-border-radius:**. Diese Eigenschaften werden in anderen Browsern dann natürlich ignoriert.

From:
<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_01:3_04_01_03



Last update: **2019/11/17 15:57**

3.4.1.4) Größenangaben

Für die Angabe von Schriftgrößen, Längen und Breiten, etc. gibt es in CSS verschiedene Einheiten.

relative Angaben

Zu bevorzugen, da die Ausgangsgröße vom Browser/Betriebssystem bestimmt werden kann, und somit die Einstellungen des Users nicht überschrieben werden.

```
font-size: 0.9em; /* Das 0.9-fache der Normalgröße */  
font-size: 15px; /* Pixel/Bildpunkt */  
font-size: 2ex; /* Angabe der Schriftgröße, relativ zum x der Normalgröße. */  
font-size: 200%; /* doppelte Größe */
```

absolut

```
font-size: 10pt; /* Punkt (1 pt entspricht 1/72 Zoll) */  
  
font-size: 1cm;  
font-size: 12mm;  
font-size: 1.5in; /* Zoll */  
font-size: 12pc; /* pica (1 pc entspricht 12pt) */
```

Weitere Informationen

Eine vollständige Auflistung der Einheiten und weitere Informationen über diese findet man unter <http://www.css4you.de/einheiten.html>.

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_01:3_04_01_04

Last update: 2019/11/17 15:57



3.4.1.5) Block- und Inlineelemente

(X)HTML-Elemente unterscheidet man in 2 Klassen: Block-Element und Inline-Elemente. Zwischen diesen beiden Typen gibt es Unterschiede in der Handhabung per CSS und den verfügbaren CSS-Eigenschaften. Die Unterschiede und Eigenschaften sind häufig auch von den Dokumenttyp-Definitionen abhängig.

Block-Elemente

- Starten in neuer Zeile und enden mit Zeilenumbruch
- Folglich ist es ohne Einsatz von CSS nicht möglich, mehrere Block-Elemente nebeneinander darzustellen
- Block-Elemente können Inline-Elemente und andere Block-Elemente beinhalten, manchmal auch nur Inline-Elemente (z.B. `<p>...</p>` - Absatz), nur Block-Elemente oder keine weiteren Elemente (→ CSS4You.de Referenz!)
- Verwendung oft als Container, also übergeordnetes Element für andere Elemente

Wichtige CSS-Eigenschaften, die nur auf Block-Elemente angewendet werden können (Erklärung zur Anwendung kommt erst später, passt hier aber am besten):

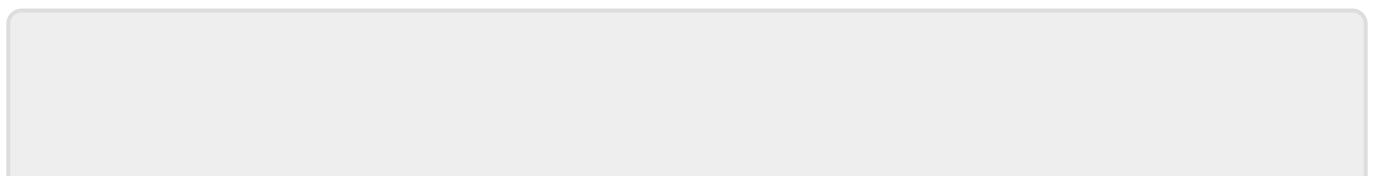
Eigenschaft	Erklärung
clear	Umfließen beenden
text-align	Textausrichtung/Horizontale Ausrichtung
text-indent	Texteinrückung

Inline-Elemente

- Werden in der aktuellen Zeile (im Textfluss) dargestellt (`
` für Zeilenumbruch davor und danach notwendig)
- Inline-Elemente können nur andere Inline-Elemente beinhalten, manchmal auch keine weiteren Elemente (→ CSS4You Referenz!)
- Verwendungszweck sind für gewöhnlich Inhalte, die der Benutzer sieht, also Text, Bilder, Links, ...

Die **vertical-align**: (Vertikale AUrichtung) - Eigenschaft kann nur auf Inline-Elemente (und Tabellenzellen) angewendet werden.

Bei vielen Elementen kann man die Darstellung als Block-Element oder Inline-Element erzwingen, auch wenn es sich nicht um ein solches Element handelt. Dies geht mit der Eigenschaft `display`: - Informationen darüber [hier](#).



From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_01:3_04_01_05



Last update: **2019/11/17 15:57**

3.4.1.6) Dokumenttyp-Definitonen

Zwar nicht unbedingt im Zusammenhang mit CSS stehend, aber generell für die Entwicklung von (X)HTML-Seiten wichtig sind Dokumenttypen (DTDs).

Was sind Dokumenttypen?

Dokumenttypen beschreiben Regeln für ein XML- oder SGML-Dokument, also sowohl, wie diese aufgebaut sein sollen (vom Entwickler), als auch wie sie vom „übersetzenden“ Programm verstanden werden soll (im Falle von HTML/XHTML ist dies der Browser). Grundsätzlich sollte man immer eine DTD angeben, tut man das nicht, verwendet der Browser eine Standarddefinition.

Beispiele für DTDs

DTD von XHTML 1.0 Transitional:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

DTD von XHTML 1.0 Frameset:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

DTD von HTML 4.01 Transitional:

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

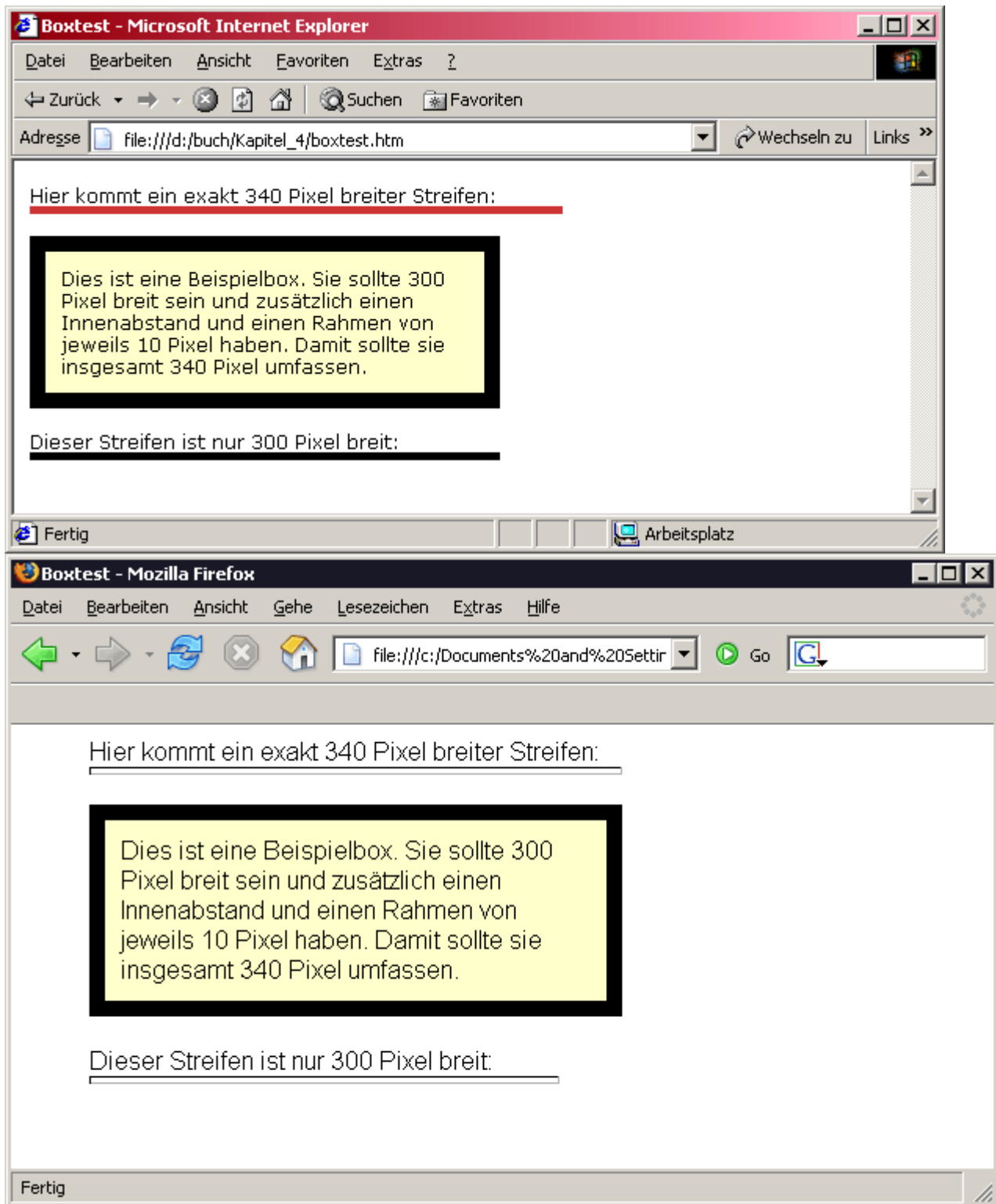
Wichtigkeit der Dokumenttyp-Angabe

Oft ist nicht ganz klar, wofür die Dokumenttyp-Angaben überhaupt notwendig sind, und es werden dann falsche oder gar keine angegeben. Diese sind jedoch sehr wichtig: Jeder Browser hat verschiedene Modis, mit denen er XHTML, HTML und CSS-Code interpretiert - und die DTDs entscheiden, welchen Modus der Browser verwendet!

- Gibt man einen anderen DTD an, als den, auf dem die eigene Webseite basiert (Beispiel: DTD-Angabe ist HTML4, man arbeitet aber mit XHTML) interpretiert der Browser den XHTML-Code als HTML4-Code, wenn es nun bei einem Element Unterschiede zwischen XHTML und HTML4 gibt, zeigt der Browser das Element eventuell nicht wie gewünscht an.
- Gibt man gar keinen oder einen sehr alten (z.B. HTML2, HTML3) DTD an, fallen die Browser in den sogenannten Quirks-Mode zurück: Die Webseite wird dann mit veralteten Versionen der W3C-Empfehlungen bzw. nicht korrekt umgesetzten W3C-Empfehlungen angezeigt. Der Grund

für das vorhandensein eines solchen Quirks-Modus liegt darin, dass auch alte Webseiten noch korrekt angezeigt werden können.

- Der Internet Explorer stellte bis Version 6 das Boxmodell falsch dar (er berechnete die Außenabstände im Zusammenhang mit der Gesamtbreite/höhe eines Elements falsch), während das bei aktuellen Browserversionen und korrektem DTD kein Problem mehr darstellt, wird genau diese falsche Berechnungsweise im Quirks-Modus des Internet Explorers und auch anderer Browser verwendet.



(Quelle der Screenshots:

<http://www.dma.ufg.ac.at/app/link/Grundlagen%3AInternet.Web.Multimedia/module/15459?step=all>)

Darum: Immer den korrekten (X)HTML-DTD angeben!

Weitere Informationen

Weitere Informationen über DTDs und deren Aufbau, wenn man selbst DTDs erstellen will, gibt es bei [Wikipedia](#).

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_01:3_04_01_06



Last update: **2019/11/17 15:58**

3.4.1.7) Exkurs: Unterschiede zwischen HTML und XHTML

Der Nachfolger von HTML 4.01 ist XHTML 1.0 (eXtensible HTML) - damit änderte sich die Basis von HTML grundlegend, auch wenn die Art und Weise, auf die es verwendet wird, nach wie vor sehr ähnlich ist. Die Basis von HTML ist SGML, die von XHTML ist XML.

Dokumenttypen

Um dem Browser klarzumachen, dass es sich um ein XHTML-Dokument handelt, müssen bestimmte Dokumenttypen verwendet werden. Man unterteilt XHTML in 3 verschiedene Typen: Strict, Transitional, Frameset - Im Normalfall verwendet man Transitional, Frameset nur, wenn man Frames verwendet. Strict hingegen macht häufig zu strikte Vorschriften und streicht auch sinnvolle Tags und Attribute (z.B. das Öffnen eines Links in einem neuen Fenster).

Die Dokumenttypangabe von XHTML Transitional ist:

```
<!DOCTYPE html PUBLIC
"-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Namensraumangabe

Aufgrund des Umstiegs auf XML muss zusätzlich der XML-Namespace mit der verwendeten Sprache im <html>-Tag angegeben werden. Das sieht so aus:

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">
```

Kleinschreibung

Während es dem Entwickler in HTML 4.01 noch freigestellt war, „müssen“ (um ein valides Dokument zu erhalten - funktionieren wird es auch anders) in XHTML sämtliche Tags und Attribute kleingeschrieben werden.

End-Tag notwendig

XHTML verlangt, dass jeder Tag auch einen End-Tag hat. Während man bei HTML 4.01 in Listen bei den einzelnen Punkten oft nur ... geschrieben hat, also auf den schließenden Tag verzichtet hat, erfordert XHTML nun auch den End-Tag, also

Eine Element, welches keinen Inhalt zwischen den Tags hat, wie z.B. der -Tag, kann auch direkt

mit einem Schrägstrich am Ende geschlossen werden: ``.

Dies gilt auch für „leere Elemente“, am wichtigsten ist hier `
`, welches nun den schließenden Tag hat.

Schachtelung

Die Elemente müssen in XHTML korrekt verschachtelt werden, d.h., ein Element, welches innerhalb eines anderen Elements geöffnet wurde, muss auch wieder innerhalb von diesem Element geschlossen werden.

- Falsch: `<p>...</p>`
- Richtig: `<p>...</p>`

Attribute müssen Werte haben

Attribute „müssen“ in XHTML einen Wert zugewiesen bekommen, die reine Angabe eines Attributs wie bei einer Checkbox das Attribut „checked“ ist nicht mehr erlaubt - stattdessen muss `checked=„checked“` verwendet werden.

id- statt name-Attribut

Während man in HTML das name-Attribut verwendete, um ein Element eindeutig zu identifizieren (z.B. für die Verwendung mit Javascript), wird dieses in XHTML mit dem Attribut `id` ersetzt.

Einbinden von CSS/JS/Sonstigem im <head>-Bereich

CSS-Definitionen, JavaScript-Code, etc. müssen in XHTML in besondere „CDATA-Bereiche“ gesteckt werden, um sicherzugehen, dass der Browser den Inhalt nicht falsch versteht.

```
<style type="text/css">
/*<![CDATA[*/
CSS-CODE
/*]]*/
</style>
```

Ja, das ist kompliziert, und die Chance, sich dieses Konstrukt fehlerfrei zu merken ist eher gering. Darum empfiehlt es sich, es zu speichern und bei Bedarf einzufügen.

Konvertierung, Validierung

- Mithilfe des W3C-Tools „Tidy“ kann man HTML-Dokumente in XHTML konvertieren. Es steht zur

Verfügung unter <http://www.w3.org/People/Raggett/tidy>

- Mit dem XHTML-Validator unter <http://validator.w3.org> kann man überprüfen, ob man ein W3-Konformes XHTML-Dokument erstellt hat.

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_01:3_04_01_07



Last update: **2019/11/17 15:58**

3.4.2) Deklarationen

- [3.4.2.1\) Die verschiedenen Selektoren](#)
- [3.4.2.2\) Klassen und IDs](#)
- [3.4.2.3\) Kommentare](#)
- [3.4.2.4\) Verschachtelung](#)
- [3.4.2.5\) Kurzformen](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_02



Last update: **2019/11/17 15:44**

11.2.2.1) Selektoren für HTML-Tags

Elementtypen

Um ein zentrales Format für alle HTML-Elemente eines bestimmten Typs zu definieren, wird als sog. **Selektor** der Name des HTML-Elementtyps angegeben, und zwar ohne spitze Klammern. Im folgenden Beispiel werden body (Dokument), h1 (Überschriften 1. Ordnung), p (Textabsätze) und li (Listeneinträge) auf diese Weise notiert.

Über den Universalselektor * hat man die Möglichkeit, Eigenschaften für alle Elemente zu definieren, wobei jedem Element natürlich nur die jeweils zulässigen Eigenschaften zugewiesen werden.

Wenn ein Format für mehrere HTML-Elementtypen definiert werden soll, werden alle gewünschten Elementtypen angegeben-

```
/* für einzelne Tags */
body { background-color: #FFFFCC;
      margin-left: 100px; }
h1 { font-size: 300%;
     color: #FF0000;
     font-style: italic;
     border-bottom: solid thin black; }

/* für mehrere Tags */
p,li { font-size: 110%;
      line-height: 140%;
      font-family: Helvetica,Arial,sans-serif;
      letter-spacing: 0.1em;
      word-spacing: 0.3em; }

/* für alle Tags */
* { color:blue; }
```

Nachfahren-Selektor

Selektor1 Selektor2 ... {Eigenschaft: Wert}

```
<style type="text/css">
  p b { color: #a00; }
</style>
```

Kind-Selektor

Selektor1>Selektor2 {Eigenschaft: Wert}

```
<style type="text/css">
  p>b { background-color: #eee; }
</style>
```

spezielle Selektoren

Übersicht

Übersicht (zur Verknüpfung und Verschachtelung)

Wichtig sind nur die für die [Links](#)

```
a:link {color:#FF0000;} /* unvisited link */
a:visited {color:#00FF00;} /* visited link */
a:hover {color:#FF00FF;} /* mouse over link */
a:active {color:#0000FF;} /* selected link */
```

(Die Reihenfolge der Deklaration ist wichtig!)

:active	aktiver Hyperlink
:after	Nach einem Element
:before	Vor einem Element
:first-child	Das erste „Kind“
:first-letter	Das erste Zeichen einer Zeile
:first-line	Die erste Zeile in einem Absatz
:focus	Fokussierte Formularelemente
:hover	mouse over
:link	Hyperlink
:visited	besuchter Link

Beispiel

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_02:3_04_02_01

Last update: **2019/11/17 15:59**



11.2.2.2) Klassen & IDs

Zur Verknüpfung von CSS mit (X)HTML werden primär sogenannte Klassen und IDs verwendet, womit einzelne und Gruppen von Elementen gestaltet werden können.

- **IDs:**

- weist man einem einzigen Element im XHTML-Code zu, es darf nur einmal vorkommen und bestimmt ein Element somit eindeutig (was auch bei Javascript wichtig ist).
- werden im XHTML-Code mit dem Attribut **id** angegeben.
- im CSS-Code werden sie mit dem Rautezeichen (#) angesteuert.
- Verwendung z.B. für ein DIV, in dem der Footer der Website liegt - dieser kommt nur einmalig vor

- **Klassen:**

- werden bzw. können mehreren Elementen im XHTML-Code zugewiesen werden
- werden im XHTML-Code mit dem Attribut **class** angegeben.
- im CSS-Code werden sie mit einem Punkt (.) angesteuert
- Verwendung z.B. für Überschriften, eine Überschrift des gleichen Ranges (z.B. h3) kann auf der selben Seite mehrmals vorkommen

Deklaration

... in (X)HTML als Attribut

```
<ul id="menueliste">
<li class="menuepunkt">eins</li>
<li class="menuepunkt">zwei</li>
<li class="menuepunkt">drei</li>
<li class="menuepunkt">vier</li>
<li class="menuepunkt">...</li>
</ul>
```

... in CSS

```
#menueliste {
/* CSS-Eigenschaften für Elemente mit ID menueliste */
}

.menuepunkt {
/* CSS-Eigenschaften für Elemente mit Klasse menuepunkt*/
}
```

auch möglich:

```
ul#menueliste {
/* CSS-Eigenschaften */
}
```



```

}

li.menuepunkt {
/* CSS-Eigenschaften */
}

```

Beispiel

```

p.merksatz {
    font-style: italic;
    border: 4px solid #EE0000;
}

/* hätte man auch ohne ID-Tag ansteuern können, da h1 ohnehin nur einmal
vorkommt */
h1#ueberschrift {
    font-size: 0.9em;
    color: darkred;
}

```

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">
<head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
    <meta http-equiv="Content-Style-Type" content="text/css" />
    <meta http-equiv="content-language" content="de" />
    <title>TITEL</title>
    <link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>

<h1 id="ueberschrift">Merksatz</h1>
<p>Text im Absatz ohne CSS-Formatierung</p>
<p class="merksatz">Das ist ein Merksatz, der per CSS als solcher Formatiert
wird. Man sollte ihn nie vergessen.</p>
</body>
</html>

```

DIV und SPAN

Die beiden HTML-Elemente `<div>` und `` haben besondere Bedeutung für Stylesheets. Der Grund ist, dass sie selber eigenschaftslos sind. Der einzige Unterschied besteht darin, dass das `div`-Element eine neue Zeile im Textfluss erzwingt, während `span` innerhalb eines Textes verwendet werden kann und keinen neuen Absatz erzeugt. Unter Verwendung von `div` oder `span` ist es also möglich, beliebigen Textabschnitten Klassenstyles zuzuweisen.

Links

- [Klassen/Klassen-Selektoren auf 4webmaster.de](#)
- [IDs/ID-Selektoren auf 4webmaster.de](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_02:3_04_02_02



Last update: **2019/11/17 15:59**

11.2.2.3) Kommentare

Kommentare sind in CSS zwischen

```
/* und */
```

möglich.

einzeilige Kommentare

```
p.left {  
    background-color: gray;    /* Die Hintergrundfarbe ist grau */  
}
```

mehrzeilige Kommentare

Das gleiche gilt für einen **mehrzeiligen Kommentar**:

```
p.left {  
    /* Das wird ein mehrzeiliger Kommentar.  
       Die Hintergrundfarbe wird jetzt als grau festgelegt. */  
    background-color: gray;  
}
```

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_02:3_04_02_03

Last update: **2019/11/17 15:59**



11.2.2.4) Verschachtelung

Da der (X)HTML-Code eigentlich nie nur eine Ebene hat, ist es möglich die CSS-Definitionen entsprechend dem (X)HTML-Code zu verschachteln. Diese Verschachtelung ist beliebig weit nach „innen“ möglich. Dazu ein kleines Beispiel:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">
<head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
  <meta http-equiv="Content-Style-Type" content="text/css" />
  <meta http-equiv="content-language" content="de" />
  <title>TITEL</title>
  <link rel="stylesheet" type="text/css" href="style.css" />

</head>
<body>
<h1>Ein <span>teilweise kursiver</span>Header</h1>
<div id="beispiel">
  Das ist ein Text direkt im DIV-Element
  <p>Und dieser Text ist im P-Element innerhalb des DIV-Elements. <span>Und
ein SPAN-Element im P-Element im DIV-Element.</span></p>
</div>
<p>Ein P-Element außerhalb des DIV-Elements - die CSS-Eigenschaft von vorher
greift hier nicht.</p>
</body>
</html>
```

style.css:

```
h1 span {
  font-style: italic;
}

div#beispiel p {
  font-weight: bold;
}

div#beispiel p span {
  text-decoration: underline;
}
```

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_02:3_04_02_04



Last update: **2019/11/17 15:59**

11.2.2.5) Kurzformen

Manche zusammengehörende Eigenschaften lassen sich als Kurzform als eine einzige Eigenschaft darstellen. Dies macht den CSS-Code kürzer und, wenn man die Kurzformen versteht, auch übersichtlicher. Man gibt die einzelnen Werte einer Kurzform mit Leerzeichen getrennt an, also:

- kurzformeigenschaft: wert1 wert2 wert3;

Dazwischen können Werte ausgelassen werden, die Reihenfolge darf jedoch nicht verändert werden. kurzformeigenschaft: wert1 wert3 wäre also ok - kurzformeigenschaft: wert3 wert2 wert1 aber nicht.

Wichtige Kurzformen

Kurzformbezeichnung	Werte	Beispiel
background	background-color background-image background-attachment background-repeat background-position	background: white url(bild.png) fixed no-repeat top left;
font	font-size line-height font-weight font-style font-variant font-family	font: italic small-caps bold 1em/1.5em verdana, sans-serif;
border	border-width border-color border-style	border: 3px black solid;
list-style	list-style-type list-style-image list-style-position	list-style: square outside;

Alle Kurzformen sind auch in der CSS4You-Referenz zu finden.

From:
<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_02:3_04_02_05



Last update: 2019/11/17 15:59

3.4.3) Vererbung & Kaskadierung

- [3.4.3.1\) Vererbung](#)
- [3.4.3.2\) Kaskadierung und Spezifität](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_03



Last update: **2019/11/17 15:46**

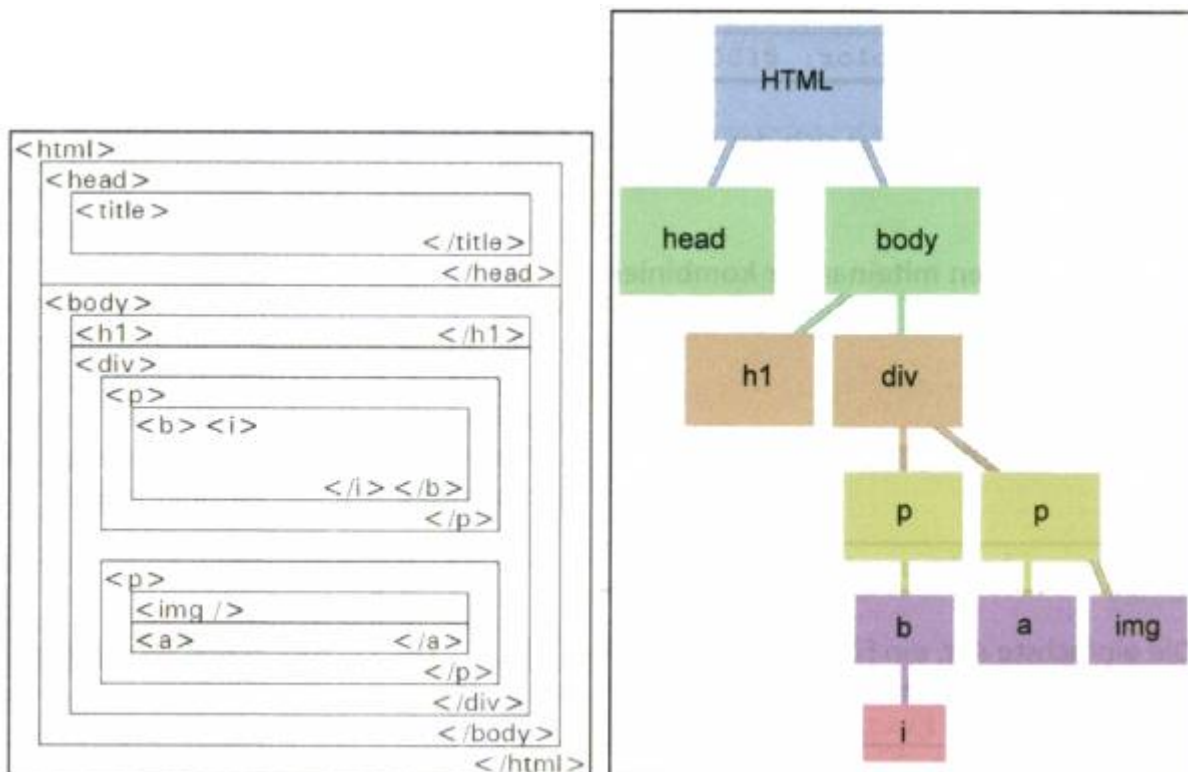
11.2.3.1) Vererbung

Vererbung heißt, dass Stileigenschaften von einem Element an ein beliebiges in ihm enthaltenes Element weitergegeben werden. Bei guter Planung kann die Vererbung Stilangaben sehr effizient machen.

Oder einfacher: Die Eigenschaften eines Elements werden von dem übergeordneten bestimmt, sofern sie nicht überschrieben werden.

Vererbung an untergeordnete Elemente

CSS-Eigenschaften werden teilweise an untergeordnete Elemente vererbt und teilweise nicht.



Eigenschaften wie die Schriftgröße, die Vorder- und Hintergrundfarben, font-weight, text-decoration, font-style, ... werden an die untergeordneten Elemente weitervererbt, d.h. von diesen übernommen, solange:

- das untergeordnete Element diese Eigenschaft nicht selbst für sich festlegt
- das untergeordnete Element für diese Eigenschaft keinen Standardwert vom Browser erhält (z.B. bei den h-Elementen die Größe und bei Links die Farbe und Unterstreichung)

Im 2. Fall kann man ein Element dazu zwingen, den Wert des übergeordneten Elements dennoch auf sich zu vererben und den Standardwert zu ignorieren, das geht indem man der Eigenschaft den Wert **inherit** zuweist. Ob dieser Wert zur Verfügung steht, ist der CSS4You-Seite der einzelnen Eigenschaften zu übernehmen.

Beispiel: font-size: inherit;

Ein Beispiel für eine Eigenschaft, die nicht vererbt wird, ist der Rahmen (border) - was durchaus logisch und gut erscheint. Die Angabe, ob ein Element vererbbar ist oder nicht, findet man stets auf der jeweiligen Eigenschaftsinfoseite auf CSS4You.

kleines Anwendungsbeispiel:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">
<head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
  <meta http-equiv="Content-Style-Type" content="text/css" />
  <meta http-equiv="content-language" content="de" />
  <title>Anwendungsbeispiel Vererbung</title>

  <style type="text/css">
/* <![CDATA[ */
div#box {
  font-size: 0.8em;
  color: white;
  background-color: black;
  font-family: Verdana; sans-serif;
}

a {
  color: inherit;
}
/* ]]> */
</style>
</head>
<body>

<div id="box">
<h1>Dieser Überschrift ist die font-size: Angabe des div-Elements egal, sie
belässt es bei ihrer eigenen Größe, weil wir sie auch nicht mit inherit zur
Vererbung zwingen.</h1>
<p>Dieser Absatz übernimmt die Eigenschaften des übergeordneten div-Elements
anstandslos.</p>
<a href="#">Wie die Überschrift möchte dieser Link die Standardeigenschaft
für color: (blau) gerne übernehmen. Mit inherit hindern wir ihn jedoch
daran.</a>
<p style="background-color: darkred">Dieser Absatz legt die background-
color-Eigenschaft selbst fest, darum wird sie auch nicht vererbt.</p>
<p>Alle gemeinsam übernehmen sie die Schriftart Verdana, weil kein Element
diese selbst festlegen will!</p>
</div>
</body>
</html>
```

Relative Einheiten bei der Vererbung

Eine Besonderheit bei der Vererbung gibt es bei der Vererbung von relativen Größen. Dann bezieht sich nämlich die Größe des untergeordneten Elements relativ auf das Übergeordnete. Weißt man einem übergeordneten Element also die Schriftgröße `font-size: 0.8em` zu, so bezieht sich eine Schriftgrößenangabe eines untergeordneten Elements auf diese `0.8em`.

Beispiel: Man weist einem `<p>`-Element `font-size: 0.8em` zu, einem untergeordneten ``-Element ebenfalls `0.8em`. Dann ist die Schriftgröße des ``-Elements erneut um den Faktor 0.8 verringert.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">
<head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
  <meta http-equiv="Content-Style-Type" content="text/css" />
  <meta http-equiv="content-language" content="de" />
  <title>Beispiel Vererbung relativer Einheiten</title>
</head>
<body>
  <p style="font-size: 0.8em;">Das ist ein Text im p-Element mit 0.8em<br />
    <span style="font-size: 0.8em;">Das ist ein Text im untergeordneten span-
    Element, dem ebenfalls 0.8em Schriftgröße zugewiesen sind.</span>
  </p>
</body>
</html>
```

Links

- [CSS4You: Informationen über die Vererbung](#)

From:
<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_03:3_04_03_01

Last update: **2019/11/17 16:00**



11.2.3.2) Kaskadierung und Spezifität

Kaskadierung

Stoßen mehrere Stylesheets aufeinander (z.B. die Standardeigenschaften vom Browser für Elemente wie h1 und Links zum einen, ein externes Stylesheet der Website und zusätzlich noch eine Inline-Styleangabe) so stellt sich die Frage, welche Eigenschaften sich durchsetzen, wenn 2 oder mehrere Stylesheets das Aussehen desselben Elements bestimmen wollen. Dabei gilt, dass sich die Eigenschaften durchsetzen die am nächsten zum Element stehen.

Folgende Hierarchie gibt es, umgesetzt wird immer die höchstmögliche Stufe in dieser Hierarchie:

- Default-Einstellungen des Browsers
- externe Style Sheets; werden mehrere externe Style Sheets mit einem Dokument verlinkt, haben im Konfliktfall die in der zuletzt eingelesenen CSS-Datei enthaltenen Anweisungen Vorrang;
- eingebettete (im Dokumentskopf definierte) Style Sheets;
- inline-Stilvorgaben.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">
<head>
<title>Kaskadierung</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<style type="text/css">
/*  */
h2 {
    font-weight: bold;
}

h3 {
    color: darkred;
}
/* ]]&gt; */
&lt;/style&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;h1&gt;Die wichtigste Überschrift, weil es keine weiteren Stilangaben gibt
werden die Standardeinstellungen des Browsers umgesetzt&lt;/h1&gt;
&lt;h2&gt;Die zweitwichtigste Überschrift, hier gibt es eine interne Stilangabe,
diese wird auch verwendet.&lt;/h2&gt;
&lt;h3 style="font-color: black;"&gt;Die drittwichtigste Überschrift, hier gibt
zusätzlich zu einer internen Stilangabe auch eine Inline-Angabe, also setzt
sich diese durch.&lt;/h3&gt;
&lt;p class="wichtig"&gt;Beispielabsatz&lt;/p&gt;
&lt;p class="wichtig" style="color: white"&gt;Beispielabsatz&lt;/p&gt;</pre></div><div data-bbox="67 951 333 967" data-label="Page-Footer"><p>Wiki - <a href="http://elearn.bgamstetten.ac.at/wiki/">http://elearn.bgamstetten.ac.at/wiki/</a></p></div>
```

```
</body>
</html>
```

Spezifität

Wie bereits bekannt, kann man mehrere (X)HTML-Elemente ineinander verschachteln und somit können durch die Vererbung einem Element verschiedene Werte für die gleiche CSS-Eigenschaft zugewiesen werden. Dabei gilt: Es setzt sich der Wert durch, der am spezifischsten für das jeweilige Element ist.

Beispiel:

- Es gibt eine div-Box, die für font-size den Wert 12pt zugewiesen bekommt.
- Darin gibt es zwei p-Boxen (also Absätze),
 - eine davon weist font-size den Wert 10pt zu,
 - die andere setzt diese Eigenschaft nicht.
 - Dann gilt bei der ersten p-Box die spezifischste Eigenschaft, also 10pt.
 - Bei der zweiten p-Box gilt die allgemeine Eigenschaft der übergeordneten Box (12pt), weil es keine eigene Eigenschaft dazu gibt.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">
<head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
  <meta http-equiv="Content-Style-Type" content="text/css" />
  <meta http-equiv="content-language" content="de" />
  <title>TITEL</title>

  <style type="text/css">
/* <![CDATA[ */
  div {
    font-size: 12pt;
  }
/* ]]> */
</style>
</head>
<body>
<div>
  <p style="font-size: 10pt;">Das ist der Absatz mit 10pt
Schriftgröße.</p>
  <p>Das ist der Absatz, der font-size nicht festlegt -> es gilt 12pt von
div</p>
</div>
</body>
</html>
```

Links

- [CSS4You: Informationen über die Kaskadierung](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_03:3_04_03_02



Last update: **2019/11/17 16:00**

3.4.4) XHTML-Elemente gestalten

- Schnellübersicht über alle CSS-Eigenschaften auf CSS4You.de (Mehr Infos mit Klick auf Eigenschaftsname)
- [3.4.4.1\) Schriftformatierungen](#)
- [3.4.4.2\) Listen und Aufzählungen](#)
- [3.4.4.3\) Links mit Pseudoklassen gestalten](#)
- [3.4.4.4\) Boxen und Boxmodell](#)
- [3.4.4.5\) Farben, Hintergrundfarben und Hintergrundbilder](#)
- [3.4.4.6\) Tabellen](#)
- [3.4.4.7\) Positionierung von Elementen](#)
- [3.4.4.8\) Float- und Clear-Eigenschaft](#)
- [3.4.4.9\) Formulare](#)
- [3.4.4.10\) Übungen zur Elementgestaltung](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_04



Last update: **2019/11/17 16:01**

3.4.4.1) Schriftformatierungen

Mit CSS ist es ohne Probleme möglich, jede Art von Schrift- und Absatzformatierung umzusetzen.

Die Eigenschaften können grundsätzlich auf alle Elemente angewendet werden, die Text beinhalten, also auch auf Listen, Formulare, Tabellen, ...

Schriftart

Die Schriftart wird mit der Eigenschaft **font-family**: festgelegt, diese nimmt als Wert sowohl eindeutige Schriftarten (z.B. verdana) als auch Schriftfamilien (z.B. serif) an. Eine Schriftfamilie beinhaltet mehrere Schriftarten, von der der Browser die 1. beim Benutzer verfügbare auswählt. Man kann (und i.d.R. tut man das auch) auch mehrere Schriftarten bzw. -familien angeben. Der Browser nimmt dann standardmäßig die 1., wenn die Schriftart nicht installiert ist die 2., usw.

Beispiel: font-family: Verdana, Arial, sans-serif;

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">
<title>Generische Schriftarten</title>
</head>
<body>
<p style="font-family: sans-serif">sans-serif</p>
<p style="font-family: serif">serif</p>
<p style="font-family: cursive">cursive</p>
<p style="font-family: fantasy">fantasy</p>
<p style="font-family: monospace">monospace</p>
</body>
</html>
```

Eigene Schriftart mitliefern

Zuerst definiert man mit font-face eine eigene Schriftart, die man aus der Schriftdatei lädt. Diese Schriftart kann man dann anwenden.

```
@font-face {
  font-family: Raleway; /* Dieser Name ist beliebig */
  src: url(Raleway-Thin.ttf);
}
```

```
<p style="font-family: Raleway">Raleway</p> /* hier muss der Wert von
font-family bei der font-face-Deklaration verwendet werden! */
```

Schriftfarbe

Die Schriftfarbe eines Elements wird (wie die Farbe aller Elemente) mit der CSS-Eigenschaft **color:** festgelegt.

Mögliche Werte sind wie bei HTML HEX-Codes (mit führender #) oder Farbnamen (green, darkgreen, ...)

Schriftgröße

Die Schriftgröße eines Elements wird mit der CSS-Eigenschaft **font-size:** festgelegt.

Diese Eigenschaft nimmt als Wert (wie alle Größen-/Längeneinheiten in CSS) verschiedene Einheiten an, z.B. px, em, cm, % → [mehr Informationen zu diesen Einheiten](#).

Weitere CSS-Eigenschaften

Eigenschaft	wichtige Werte	Erklärung
font-style	italic, normal	Bestimmt die Darstellung der Schrift, im wesentlich kursiv oder normal
font-weight	bold, normal	Legt die „Schriftgewichtung“ fest
text-align	left, right, center, justify	Ausrichtung des Textes
text-decoration	underline, overline, line-through, blink, none	Ermöglicht Unter/Durch/Überstreichungen und blinkenden Text
text-shadow	Farbe, 3x Länge	Informationen s. hier
word-spacing	Längenangabe	Abstand zwischen den Wörtern
letter-spacing	Längenangabe	Abstand zwischen den Buchstaben/Zeichen

Weitere Informationen auf CSS4You

Weitere Informationen zu diesen und weiteren CSS-Eigenschaften und den anwendbaren Werten gibt es unter <http://www.css4you.de/Schrifteigenschaften/fontproperty.html> (Schrifteigenschaften) und unter <http://www.css4you.de/Texteigenschaften/textproperty.html> (Texteigenschaften).

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_04:3_04_04_01

Last update: 2019/11/17 16:04



3.4.4.2) Listen und Aufzählungen

Auch die bereits bekannten Listen kann man mit CSS einfach formatieren, häufig werden auch Menüs von Webseiten mit Listen gestaltet.

Zur Wiederholung eine ungeordnete Liste in (X)HTML:

```
<ul>
  <li>Ein Punkt</li>
  <li>Ein weiterer Punkt</li>
  <li>Ein weiterer Punkt</li>
</ul>
```

Listen werden in ul (ungeordnete Listen → Aufzählungszeichen) und ol (geordnete Listen → Zahlen oder Buchstaben zur Aufzählung) getrennt.

Wichtige CSS-Eigenschaften für Listen

Eigenschaft	erlaubte Werte	Erklärung	Anwendung möglich auf
list-style-type	none (ol+ul), disc, circle, square (bei ul), decimal, lower-alpha, upper-alpha, lower-roman, upper-roman, lower-greek, decimal-leading-zero, hebrew, armenian, georgian, cjk-ideographic, hiragana, katakana, hiragana-iroha, katakana-iroha (bei ol)	Bestimmt das Aufzählungszeichen (Standardmäßig disc - gefüllter Kreis)	ul, ol, li
list-style-image	none, url(), Inherit (vererbt vom übergeordneten Element)	Für die Verwendung einer Grafik anstatt eines Symbols von list-style-type	ul, ol, li
list-style-position	inside, outside	Bestimmt, ob Aufzählungszeichen innerhalb oder außerhalb des Textblocks stehen sollen (Standard: outside)	ul, ol, li

Weitere Informationen zu den einzelnen Eigenschaften gibt es unter <http://www.css4you.de/listproperty.html>.

Natürlichen können die „allgemeinen“ CSS-Eigenschaften für Text, Links, Bilder, Farben, etc. auch innerhalb von Listen angewendet werden.

Verschachtelung

Bekanntlicherweise können Listen verschachtelt werden. In X(HTML) sieht das folgendermaßen aus:

```

<ul>
  <li>Ein Punkt</li>
  <li>Ein weiterer Punkt
    <ul>
      <li>Unterpunkt
        <ul>
          <li>Unter-Unterpunkt</li>
        </ul>
      </li>
      <li>Unterpunkt</li>
    </ul>
  </li>
  <li>Ein weiterer Punkt</li>
</ul>

```

Zur Formatierung der Unterpunkte arbeitet man mit Nachfahrenselektoren, das sieht folgendermaßen aus (CSS-Code):

```

ul {
  list-style-type: disc;
}
ul ul
{
  list-style-type: circle;
}
ul ul ul
{
  list-style-type: square;
}

```

Würde man keine Nachfahrenselektoren deklarieren, würde das übergeordnete ul-Element die Eigenschaft an die untergeordneten vererben.

BSP: Menü mit Listen

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Listen inline</title>
<style type="text/css">
/*  */
body {
  font-family: sans-serif;
  background-color: #FFF;
  color: #306;
}
ul {
</pre>
</div>
<div data-bbox="68 952 296 967" data-label="Page-Footer"><a href="http://elearn.bgamstetten.ac.at/wiki/">http://elearn.bgamstetten.ac.at/wiki/</a></div>
<div data-bbox="744 952 937 966" data-label="Page-Footer">Printed on 2020/06/02 10:37</div>
```

```
list-style-type: none;
display: inline;
padding-left: 0;
margin-left: 0;
}
li {
display: inline;
padding: 3px 15px;
border-left: 1px solid #000;
}

/* ]]> */
</style>
</head>
<body>

<ul>
  <li>Ein Punkt</li>
  <li>Ein Punkt</li>
  <li>Ein Punkt</li>
  <li>Ein Punkt</li>
</ul>

</body>
</html>
```

Links

[CSS4You: Eigenschaften für Listen](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_04:3_04_04_02



Last update: **2019/11/17 16:04**

3.4.4.3) Links mit Pseudoklassen gestalten

Für Links gibt es verschiedene Zustände. Wurde ein Link vom Besucher bereits aufgerufen, so hat er den Zustand „visited“, wird ein Link mit der Maus berührt, hat er den Zustand „hover“, usw. - Mit CSS kann man für jeden dieser Zustände bequem eine eigene Formatierung umsetzen, wie es auch bei vielen Websites gemacht wird.

Bei Pseudoelementen und Pseudoklassen notiert man zuerst das betroffene HTML-Element, im Beispiel das a-Element für Verweise. Dahinter folgt ein Doppelpunkt und dahinter eine erlaubte Angabe, im Beispiel etwa `link` (für noch nicht besuchte Verweisziele), `visited` (für bereits besuchte Verweisziele), `hover` (für Verweise, während der Anwender mit der Maus darüber fährt), `active` (für angeklickte Verweise) und `focus` (für z.B. mit der Tastatur ausgewählte Verweise).

Wichtig ist, dass dies keine frei wählbaren Namen sind, sondern feste Schlüsselwörter.

Zustandsname	Erklärung
link	Standard: Der Link wurde noch nicht aufgerufen, wird auch nicht mit der Maus berührt oder ähnliches
visited	Der Link wurde vom Besucher bereits aufgerufen, er ist im Verlauf des verwendeten Browsers
hover	Der Link wird mit der Maus gerade berührt
focus	Der Link hat aktuell den Fokus
active	Der Link wird gerade angeklickt, die Maustaste wurde aber noch nicht losgelassen

Diesen Zuständen kann mit Pseudoklassen eine eigene Formatierung zugeordnet werden. Dabei wird zur Auswahl in CSS einfach, wie bisher, das Element (und evtl. die Klasse/ID) angegeben, ergänzt mit **:zustand**

Praxisbeispiel

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Links</title>

<style type="text/css">
/*  */
body {background-color: #333; }
a:link { color: red; }
a:visited { color: green; }
a:hover { color: blue; }
a:focus { color: white; }
a:active { color: yellow; }
/* ]]&gt; */
&lt;/style&gt;
&lt;/head&gt;</pre></div><div data-bbox="67 952 296 967" data-label="Page-Footer"><a href="http://elearn.bgamstetten.ac.at/wiki/">http://elearn.bgamstetten.ac.at/wiki/</a></div><div data-bbox="744 952 936 966" data-label="Page-Footer">Printed on 2020/06/02 10:37</div>
```

```
<body>  
<a href="#">Ein bedeutsamer Link</a> <br />  
<a href="#">Ein bedeutsamer Link</a> <br />  
  
</body>  
</html>
```

Links

Pseudoelemente und -klassen

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_04:3_04_04_03

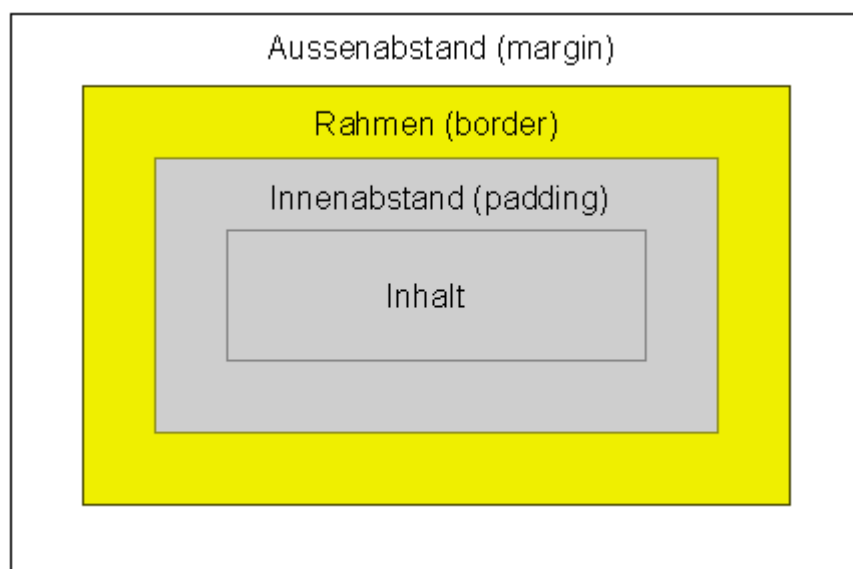


Last update: **2019/11/17 16:04**

3.4.4.4) Boxen und Boxmodell

Wie bereits erwähnt unterscheidet man Block-Elemente und Inline-Elemente. Block-Elemente verwenden zur Darstellung das Boxmodell. Eine Box (also ein Block-Element) hat immer einen Innenabstand (padding), der zwischen Inhalt und dem Rahmen ist, einen Rahmen (border) und einen Außenabstand (margin), der den Abstand rund um den Rahmen angibt.

Grafische Darstellung des Boxmodells:



(© css4you.de)

CSS-Eigenschaften

Eigenschaft	Wert	Erklärung
padding	Größenangabe	Innenabstand
margin	Größenangabe	Außenabstand
border-color	HEX-Code oder Farbbezeichnung	Farbe des Rahmens
border-width	Größenangabe	Breite des Rahmens
border-style	none, dotted, dashed, solid, double, groove, ridge, inset, outset	Style des Borders

Es ist auch möglich, eine Rahmen- oder Abstandseigenschaft nur einer Seite der Box zuzuweisen; weitere Informationen dazu und allgemein zu den Rahmen-Eigenschaften gibt es wie immer auf CSS4You: [Eigenschaften für Rahmen](#), [Eigenschaften für Innenabstand](#), [Eigenschaften für Außenabstand](#)

Umfangreiche Informationen zu Boxen und dem Boxmodell findet man unter <http://www.css4you.de/wsboxmodell/index.html>.

Beispiel

Die DIV-Box hat einen Innenabstand, Außenabstand und einen Rahmen. Innerhalb dieser DIV-Box gibt

es eine weitere DIV-Box die den Text beinhaltet. Die äußere DIV-Box muss ein Block-Element sein, die innere hätte auch ein Inline-Element sein können (z.B. span).

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">
<head>
<title>Boxmodell</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<style type="text/css">
/*  */
body {
    background-color:#FFF5E6;
    color: #995C00;
    margin: 0;
    padding: 0;
    font: 100.01% sans-serif;
}
#box {
    width:20em;
    background-color:#FFC266;
    border: 3em solid #FF9900;
    padding: 4em;
    margin: 2em;
}
#inhalt {
    background-color:#FFF5E6;
    font-size: 3em;
}
/* ]]&gt; */
&lt;/style&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;div id="box"&gt;
    &lt;div id="inhalt"&gt;Inhaltsbereich&lt;/div&gt;
&lt;/div&gt;

&lt;/body&gt;
&lt;/html&gt;</pre></div><div data-bbox="67 760 147 781" data-label="Section-Header"><h2>Links</h2></div><div data-bbox="101 809 496 879" data-label="List-Group"><ul><li>• <a href="#">CSS4You: Informationen über das Boxmodell</a></li><li>• <a href="#">CSS4You: Eigenschaften für Rahmen</a></li><li>• <a href="#">CSS4You: Eigenschaften für Außenabstand</a></li><li>• <a href="#">CSS4You: Eigenschaften für Innenabstand</a></li></ul></div><div data-bbox="67 951 333 967" data-label="Page-Footer"><p>Wiki - <a href="http://elearn.bgamstetten.ac.at/wiki/">http://elearn.bgamstetten.ac.at/wiki/</a></p></div>
```

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_04:3_04_04_04



Last update: **2019/11/17 16:04**

3.4.4.5) Farben, Hintergrundfarben und Hintergrundbilder

Farben

Jedem Element kann eine Farbe zugewiesen werden. Hier ist damit die Vordergrundfarbe, also z.B. die Farbe vom Text selbst gemeint. Die zuständige Eigenschaft ist **color:** und sie nimmt Farben im HEX-Format oder als Wort an.

Hintergrundfarben

Man kann mit CSS Hintergrundfarben für alle (X)HTML-Elemente vergeben, also sowohl eine Farbe für die gesamte Seite als auch nur für einen bestimmten Text, eine Tabellenzeile, etc.

Die Eigenschaft für die Hintergrundfarbe ist dabei immer **background-color:**, als Wert wird eine Farbe entweder im HEX-Format oder als Wort (green, ...) angenommen.

[\(X\)HTML-Beispielcode anzeigen](#)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">
<head>
<title>Hintergrundfarbe</title>
<link rel="stylesheet" type="text/css" href="style.css" />
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
<body>
<h1>Hintergrundfarbe</h1>
<p id="text">Das ist ein Absatz mit Text. Er hat eine eigene
Hintergrundfarbe. Das ist ein Absatz mit Text. Er hat eine eigene
Hintergrundfarbe. Das ist ein Absatz mit Text. Er hat eine eigene
Hintergrundfarbe. Das ist ein Absatz mit Text. Er hat eine eigene
Hintergrundfarbe. Das ist ein Absatz mit Text. Er hat eine eigene
Hintergrundfarbe. Das ist ein Absatz mit Text. Er hat eine eigene
Hintergrundfarbe. Das ist ein Absatz mit Text. Er hat eine eigene
Hintergrundfarbe.</p>

</body>
</html>
```

[Dazu gehörenden CSS-Beispielcode anzeigen](#)

```
body {  
    background-color: #FFFF99;  
}  
  
h1 {  
    font-size: 1.7em;  
}  
  
p#text {  
    background-color: darkgreen;  
    color: white;  
}
```

Hintergrundbilder

Ebenso können für alle Elemente Hintergrundbilder vergeben werden. Die Eigenschaft dazu lautet **background-image:**, als Wert wird ein Link zu einem Bild angenommen:

url(/images/hintergrundbild.png)

Zwei weitere wichtige Eigenschaften für Hintergrundbilder gibt es:

- **background-repeat:** Legt fest, ob und in welche Richtung das Hintergrundbild wiederholt werden soll. Mögliche Werte sind no-repeat, repeat (in alle Richtungen wiederholen), repeat-x (horizontal wiederholen), repeat-y (vertical wiederholen). Am sinnvollsten ist hier bei einem von oben nach unten gehenden Farbverlauf repeat-x; bei einem von links nach rechts gehendem Farbverlauf repeat-y; bei einem Hintergrundbild mit einem Muster repeat.
- **background-attachment:** Damit bestimmt man, ob das Hintergrundbild mit der Seite mitscrollen soll (Wert scroll) oder fixiert an seinem Platz stehen soll (Wert fixed)
- **background-position:** Legt die Position des Hintergrundbildes fest, je nach der background-repeat Angabe. Genauere Informationen [hier](#)

Es gibt noch weitere Eigenschaften, um Hintergrundbilder noch genauer steuern zu können, diese sind unter <http://www.css4you.de/backgroundproperty.html> zu finden.

Hinweis: Einen Farbverlauf kann man sich entweder mit Bildbearbeitungsprogrammen wie Photoshop oder mit Farbverlaufs-Generatoren im Internet (z.B. <http://www.generateit.net/gradient>) erstellen.

Beispiel:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">  
<head>  
    <meta http-equiv="content-type" content="text/html; charset=UTF-8" />  
    <meta http-equiv="Content-Style-Type" content="text/css" />  
    <meta http-equiv="content-language" content="de" />  
    <title>TITEL</title>  
  
    <style type="text/css">  
/* <![CDATA[ */
```

```
body {
  background-image: url(farbverlauf.png); /* alternative: muster */
  background-color: white;
  background-repeat: repeat-x; /* alternativen: repeat, repeat-y, no-repeat
*/
  /*background-attachment: fixed */ /* auskommentieren um Wirkung zu sehen!
*/
      /*background-position: 15px 25px */ /* auskommentieren um Wirkung
zu sehen! */
}

div#beispiel_div {
  height: 900px;
  background-color: white;
  border: black solid 3px;#
  border-radius: 10px;
  width: 400px;
  margin: 100px;
}

/* ]]> */
</style>
</head>
<body>
<div id="beispiel_div">Das ist der Inhalt der DIV-Box</div>
</body>
</html>
```

Farbverlaufs-PNG:

Links

[CSS4You: Eigenschaften zu Hintergrundbildern](#)

[CSS4You: Verfügbare Farbnamen](#)

[CSS4You: Übersicht von websicheren Farben \(in HEX-Codes\)](#)

[CSS4You: Systemfarben](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_04:3_04_04_05



Last update: **2019/11/17 16:04**

3.4.4.6) Tabellen

Auch Tabellen können mithilfe von CSS grafisch aufgehübscht werden. Dazu werden die allgemeinen Eigenschaften, wie color, border, etc. auf Elemente der Tabelle, also <table>, <tr>, <td>, ... angewendet. Mit (X)HTML werden verschiedene neue Tags für Tabellen eingeführt, praktisch werden diese jedoch kaum genutzt. Deswegen beschränken wir uns auf den <caption>-Tag, der der Tabelle einen Titel gibt.

Beispiel

index.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">
<head>
<meta http-equiv="content-type" content="text/html; charset=UTF-8" />
<title>Tabelle</title>
<link rel="stylesheet" type="text/css" href="style.css" />
</head>

<body>
<table cellpadding="0">
  <caption>Tabelle XY</caption>
  <tr><th class="ohne">&Uuml;bersicht</th> <th>Spalte 1</th> <th>Spalte
2</th></tr>
  <tr><th>Zeile 1</th><td>Erste Zelle</td><td>Zweite Zelle</td></tr>
  <tr class="gerade"><th>Zeile 2</th><td>Dritte Zelle</td><td>Vierte
Zelle</td></tr>
  <tr><th>Zeile 3</th><td>F&uuml;nfte Zelle</td><td>Sechste Zelle</td></tr>
  <tr class="gerade"><th>Zeile 4</th><td>Siebte Zelle</td><td>Achte
Zelle</td></tr>
  <tr><th>Zeile 5</th><td>Neunte Zelle</td><td>Zehnte Zelle</td></tr>
  <tr class="gerade"><th>Zeile 6</th><td>Elfte Zelle</td><td>Zw&ouml;lfte
Zelle</td></tr>
  <tr><th>Ergebnis</th><td>Erste Spalte</td><td>Zweite Spalte</td></tr>
</table>

</body>
</html>
```

style.css

```
body {
    background-color: #CFDCE6;
    font: 100.01% Verdana, sans-serif;
    color: #706348;
}
table {
    border-collapse: collapse;
}
td, th {
    padding: 10px;
    width: 15em;
    border: 1px solid #807459;
    font: 0.8em sans-serif;
}
th {
    font-weight: bold;
    letter-spacing: 0.5em;
}
caption {
    text-transform: uppercase;
    font-size: 1.5em;
    margin-bottom: 10px;
}

tr.gerade {
    background-color: #E6DFCF;
    color: #647D8F;
}
tr {
    background-color: white;
}
tr th {
    background: url(hg_table2.gif) white no-repeat 3px 3px;
}
tr.gerade th {
    background: url(hg_table1.gif) #E6DFCF no-repeat 3px 3px;
}
```

Grafiken



From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_04:3_04_04_06



Last update: **2019/11/17 16:04**

3.4.4.7) Positionieren von Elementen

Es gibt verschiedene Möglichkeiten, Elemente per CSS an eine vorbestimmte Position zu setzen.

Positionierung von Elementen

Statische Positionierung

Die statische Positionierung ist die häufigst-angewendete, und darum auch die Standardpositionierungsart. Die Elemente werden damit im normalen Textfluss dargestellt, wie bisher bei allen Beispielen.

Die CSS-Regel für die statische Positionierung ist **position: static** - weil es aber sowieso die Standardeinstellung ist, muss diese nicht angegeben werden.

(X)HTML/CSS-Beispielcode anzeigen

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">
<head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
  <meta http-equiv="Content-Style-Type" content="text/css" />
  <meta http-equiv="content-language" content="de" />
  <title>TITEL</title>

<style type="text/css">
/*  */
&lt;!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"&gt;
&lt;html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de"&gt;
&lt;head&gt;
&lt;title&gt;statische Positionierung&lt;/title&gt;
&lt;meta http-equiv="Content-Type" content="text/html; charset=UTF-8" /&gt;
&lt;style type="text/css"&gt;
/* <![CDATA[ */
div {
    width: 12em;
    color: #0000CC;
    border: dotted 1px;
    padding: 1em;
}
.box1 {
    background-color: #FFCC00;
    position:static;
}
.box2 {</pre></div><div data-bbox="67 951 333 967" data-label="Page-Footer"><p>Wiki - <a href="http://elearn.bgamstetten.ac.at/wiki/">http://elearn.bgamstetten.ac.at/wiki/</a></p></div>
```

```

        background-color: #FFFF99;
        position:static;
    }
/* ]]> */
</style>
</head>
<body>
<div class="box1">Erste Box</div>
<div class="box2">Zweite Box</div>
</body>
</html>

```

Absolute Positionierung

Bei der absoluten Positionierung werden die Elemente aus dem Textfluss genommen, und an eine festgelegte Stelle gerückt. Dadurch können sich auch mehrere Elemente überlagern.

Angegeben wird wieder Positionierungstyp mit **position: absolute**, während die Position mit den Eigenschaften **left: ..**, **top: ..**, **right: ..** und **bottom: ..** festgelegt wird. Die Werte für diese Eigenschaften können beliebige Längenangaben mit allen bekannten Einheiten sein. Sie beziehen sich, wenn es kein übergeordnetes Element gibt, das selbst mit `position: absolute;` positioniert wird, auf das Browserfenster - ansonsten auf das übergeordnete Element (aber eben nur unter der **Bedingung**, das dieses übergeordnete Element selbst absolut positioniert ist!). Meist gibt man für `left`, `top`, `right` und `bottom` einen Wert mit der Einheit `em` an, weil sich das Design somit an die Schriftgröße des Benutzers anpasst.

- Die Eigenschaften stehen immer für Abstände, also `top`: legt den Abstand vom obersten Rand des Browserfensters bzw. des übergeordneten Elements fest.
- Man verwendet immer ENTWEDER `top: ..` ODER `bottom: ..` bzw. `left: ..` ODER `right: ..`

Beispiel 1: 2 DIVs ohne übergeordnetes Element

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">
<head>
<title>Positionierung</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<style type="text/css">
/*  */
div {
    width: 12em;
    color: #00C;
    border: dotted 1px;
    padding: 1em;
}
.box1 {
    background-color: #FC0;
    position:static;
</pre>
</div>
<div data-bbox="67 952 296 967" data-label="Page-Footer"><a href="http://elearn.bgamstetten.ac.at/wiki/">http://elearn.bgamstetten.ac.at/wiki/</a></div>
<div data-bbox="744 952 928 966" data-label="Page-Footer">Printed on 2020/06/02 10:37</div>
```

```

}
.box2 {
    background-color: #FF9;
    position: absolute;
    left: 4em;
    top: 2em;
}
/* ]]> */
</style>
</head>
<body>

<div class="box1">Erste Box</div>
<div class="box2">Zweite Box</div>

</body>
</html>

```

Beispiel 2: 2 DIVs ohne übergeordnetes Element, 1 DIV das einem der beiden DIVs untergeordnet ist

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">
<head>
<title>Positionierung</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<style type="text/css">
/*  */
div {
    width: 12em;
    color: #00C;
    border: dotted 1px;
    padding: 1em;
}
.box1 {
    background-color: #FC0;
    position: static;
}
.box2 {
    background-color: #FF9;
    position: absolute;
    left: 4em;
    top: 2em;
}
.box3 {
    background-color: #CCF;
    position: absolute;
    left: 4em;
    top: 2em;
}
</pre>
</div>
<div data-bbox="68 951 333 967" data-label="Page-Footer">Wiki - <a href="http://elearn.bgamstetten.ac.at/wiki/">http://elearn.bgamstetten.ac.at/wiki/</a></div>
```

```

/* ]]> */
</style>
</head>
<body>
<div class="box1">Erste Box
</div>
<div class="box2">Zweite Box
    <div class="box3">Dritte Box</div>
</div>

</body>
</html>

```

An Beispiel 2 sieht man: Wäre der Bezugspunkt immer das Browserfenster, und nicht das übergeordnete Element, müsste die 3. Box an der gleichen Position wie die 2. Box sein.

Beispiel 3: gleich wie BSP2, jedoch wird die übergeordnete Box nicht mehr mit position: absolute positioniert

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">
<head>
<title>Positionierung</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<style type="text/css">
/* <![CDATA[ */

div {
    width: 12em;
    color: #00C;
    border: dotted 1px;
    padding: 1em;
}
.box1 {
    background-color: #FC0;
    position: static;
}
.box2 {
    background-color: #FF9;
}
.box3 {
    background-color: #CCF;
    position: absolute;
    left: 4em;
    top: 2em;
}

```

```

/* ]]> */
</style>
</head>
<body>
<div class="box1">Erste Box
</div>
<div class="box2">Zweite Box
    <div class="box3">Dritte Box</div>
</div>

</body>
</html>

```

Beispiel 3 zeigt die Bedingung für die Beziehung zwischen übergeordnetem und untergeordnetem Element.

Relative Positionierung

Bei der relativen Positionierung wird ein Element relativ zu ihrem eigentlichen (unsprünglichen) Platz (im Textfluss) positioniert. Dabei verändert sich die Position der umgebenden Elemente nicht.

[\(X\)HTML/CSS-Beispielcode anzeigen](#)

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">
<head>
<title>Positionierung</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<style type="text/css">
/* <![CDATA[ */

div {
    width: 12em;
    color: #00C;
    border: dotted 1px;
    padding: 1em;
}
.box1 {
    background-color: #FC0;
    position: static;
}
.box2 {
    background-color: #FF9;
    position: relative;
    left: 4em;
    top: 2em;
}

```

```

}

/* ]]> */
</style>
</head>
<body>
<div class="box1">Erste Box</div>
<div class="box2">Zweite Box</div>
<div class="box3">Dritte Box</div>

</body>
</html>

```

Fixierte Positionierung

Die fixierte Positionierung funktioniert wie die absolute, mit 2 Unterschieden:

- Die Positionsangaben beziehen sich IMMER auf das Browserfenster, nicht auf eventuelle übergeordnete Elemente
- Das Element scrollt mit den übrigen Elementen nicht mit, es bleibt immer an dieser einen Stelle.

[\(X\)HTML/CSS-Beispielcode anzeigen](#)

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">
<head>
<title>Positionierung</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<style type="text/css">
/*  */
body {
    font: 100.01% sans-serif;
    color: #00C;
    background-color: #fff;
}
div {
    width: 12em;
    border: dotted 1px;
    padding: 1em;
}
.box1 {
    background-color: #FFCC00;
    position: static;
}
.box2 {
</pre>
</div>
<div data-bbox="67 952 296 967" data-label="Page-Footer">
<a href="http://elearn.bgamstetten.ac.at/wiki/">http://elearn.bgamstetten.ac.at/wiki/</a>
</div>
<div data-bbox="744 952 936 966" data-label="Page-Footer">Printed on 2020/06/02 10:37</div>
```

```
        background-color: #FFFF99;
        position: fixed;
        left: 4em;
        top: 2em;
    }
/* ]]> */
</style>
</head>
<body>
<div class="box1">Erste Box</div>
<div class="box2">Zweite Box</div>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam
nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat
volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation
ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.</p>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam
nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat
volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation
ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.</p>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam
nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat
volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation
ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.</p>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam
nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat
volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation
ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.</p>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam
nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat
volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation
ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.</p>
</body>
</html>
```

Überlappungen (z-Achse) und Durchsichtigkeit

Überlappungen

Wie bereits gesehen kann es bei manchen Positionierungstypen zu Überlappungen kommen. Will man festlegen, welche(s) Element(e) dabei vorne liegen, macht man das mit der Eigenschaft **z-index**: - je höher der Wert dieser Eigenschaft, desto weiter vorne erscheint das Element. Der Wert muss in keiner bestimmten Folge angegeben werden, sondern es können beispielsweise auch die Werte 5, 16, 33, 67 vorkommen. z-index funktioniert nur bei absolut positionierten Elementen.

[\(X\)HTML/CSS-Beispielcode anzeigen](#)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">
<head>
<title>z-index</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<style type="text/css">
/*  */
div {
    width: 12em;
    color: #00C;
    border: dotted 1px;
    padding: 1em;
}
.box1 {
    background-color: #FC0;
    position: absolute;
    left: 0;
    top: 1em;
    z-index: 3;
}
.box2 {
    background-color: #FF9;
    position: absolute;
    left: 4em;
    top: 2em;
    z-index: 1;
}
.box3 {
    background-color: #FFF;
    position: absolute;
    left: 6em;
    top: 3em;
    z-index: 2;
}
/* ]]&gt; */
&lt;/style&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;div class="box1"&gt;Erste Box&lt;/div&gt;
&lt;div class="box2"&gt;Zweite Box&lt;/div&gt;
&lt;div class="box3"&gt;Dritte Box&lt;/div&gt;
&lt;/body&gt;
&lt;/html&gt;</pre></div><div data-bbox="67 806 361 827" data-label="Section-Header"><h2>Durchsichtigkeit (Opacity)</h2></div><div data-bbox="67 853 900 907" data-label="Text"><p>Ebenfalls interessant ist die Eigenschaft <b>opacity</b>, diese gibt den Durchsichtigkeitsgrad eines Elements an. Leider wird diese derzeit noch von vielen Browser unterschiedlich implementiert. Als Wert für opacity: gibt man einen Wert zwischen 0 und 1 an.</p></div><div data-bbox="68 951 296 967" data-label="Page-Footer"><a href="http://elearn.bgamstetten.ac.at/wiki/">http://elearn.bgamstetten.ac.at/wiki/</a></div><div data-bbox="744 951 937 966" data-label="Page-Footer">Printed on 2020/06/02 10:37</div>
```


From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_04:3_04_04_07



Last update: **2019/12/01 22:17**

3.4.4.8) Float- und Clear-Eigenschaft

Neben den verschiedenen Positionierungsarten gibt es eine weitere Möglichkeit, die Darstellung von Webseiten zu verändern.

Float: Umfließen

Float setzt die für Block-Elemente geltende Regel, dass diese in einer neuen Zeile beginnen außer Kraft. Es bewirkt also, dass keine Veränderung des normalen Textflusses stattfindet.

Das geschieht, indem das betroffene Element von den anderen umflossen wird. Dem Element, das umflossen werden soll, weist man die Regel float: left (für links umfließen) oder float: right (für rechts umfließen) zu. Der Standardwert ist float: none, also nicht umfließen.

Folgendes **Beispiel** zeigt ein Bild, das von einem Text im Absatz umflossen werden soll:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">
<head>
<title>Float</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<style type="text/css">
/*  */
* {
    padding: 0;
    margin: 0;
}
body {
    font: 100.01% sans-serif;
}
div {
    background-color: #EBE6E0;
    margin: 10px;
}
.rechts {
    float: right;
}
.links {
    float: left;
}

/* ]]&gt; */
&lt;/style&gt;</pre></div><div data-bbox="67 951 296 967" data-label="Page-Footer"><a href="http://elearn.bgamstetten.ac.at/wiki/">http://elearn.bgamstetten.ac.at/wiki/</a></div><div data-bbox="742 951 937 966" data-label="Page-Footer">Printed on 2020/06/02 10:37</div>
```

```

</head>
<body>
<div><p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam
nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat
volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation
ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo
consequat.</p></div>
<hr />
<div><p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam
nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat
volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation
ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo
consequat.</p></div>

</body>
</html>

```



Bild:

Clear: Umfließen beenden

Das Problem ist nun, dass das Bild nicht nur von dem einen Absatz umflossen wird, sondern von allen folgenden Elementen. Das Umfließen muss also nach dem gewünschten Element wieder mit der Regel `clear: left`, `clear: right` oder `clear: both` stoppen. Die `clear`-Regel muss dem Element zugewiesen werden, die das umflossene Element als letztes umfließen soll.

Nun das gleiche **Beispiel** wie oben, nur mit der Anwendung von `clear` – Es muss nur der CSS-Code geändert werden:

```

<style type="text/css">
/*  */
* {
    padding: 0;
    margin: 0;
}
body {
    font: 100.01% sans-serif;
}
.links {
    float: left;
    padding-right: 20px;
}
.rechts {
    float: right;
}
</pre>
</div>
<div data-bbox="67 951 333 967" data-label="Page-Footer">Wiki - <a href="http://elearn.bgamstetten.ac.at/wiki/">http://elearn.bgamstetten.ac.at/wiki/</a></div>
```

```
padding-left: 20px;
}
div {
background-color: #EBE6E0;
margin: 10px;
}
hr {
clear: left;
}
/* ]]> */
</style>
```

Das Problem mit den unterschiedlichen Höhen

Wie man im BSP sieht, gibt es noch ein Problem: Der Absatz wurde durch den DIV-Block grau hinterlegt, allerdings nicht so hoch wie das Bild, obwohl es im selben DIV liegt. Um das Problem zu beseitigen gibt es einen einfachen Trick: `overflow: hidden` – dem DIV muss man dazu jedoch eine fixe Breite zuweisen.

Beispiel vollendet - Es muss nur der CSS-Code geändert werden:

```
<style type="text/css">
/*  */
* {
padding: 0;
margin: 0;
}
body {
font: 100.01% sans-serif;
}
.links {
float: left;
padding-right: 20px;
}
.rechts {
float: right;
padding-left: 20px;
}
div {
background-color: #EBE6E0;
margin: 10px;
overflow: hidden;
width: 900px;
}
hr {
clear: left;
}
/* ]]&gt; */
&lt;/style&gt;</pre></div><div data-bbox="68 952 297 967" data-label="Page-Footer"><a href="http://elearn.bgamstetten.ac.at/wiki/">http://elearn.bgamstetten.ac.at/wiki/</a></div><div data-bbox="744 952 937 967" data-label="Page-Footer">Printed on 2020/06/02 10:37</div>
```

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_04:3_04_04_08



Last update: **2019/11/17 16:05**

3.4.4.9) Formulare

XHTML-Tags für Formulare

Formulare haben mit XHTML einige neue Tags bekommen, um sie genauer bzw. bequemer formatieren und ansteuern zu können.

Tag	wichtige Attribute	Bedeutung
<form>...</form>	action, method	Innerhalb der Tags befindet sich das Formular.
		action: Zielfeld des Formulars beim Absenden, method: Methode der Datenübermittlung (POST oder GET)
<fieldset>...</fieldset>	-	Fasst verschiedene Felder eines Formulars zu einem Block zusammen
<legend>...</legend>	-	Gibt dem <fieldset> eine Bezeichnung, die angezeigt wird
<label>...</label>	for	Darin wird zur besseren Verwendung, u.a. mit CSS, die für den Nutzer sichtbare Bezeichnung des Feldes geschrieben.
		for: Muss dem Attributwert „id“ des input-Felds entsprechen
<input />	type, name, id, size	Eingabeelement
		type: Feldtyp (z.B. text), name: eindeutige Bezeichnung für PHP, id: eindeutige Bezeichnung für HTML/CSS/JS, size: Größe des Feldes

Gestaltung mit CSS

Durch die Verwendung der oben genannten Tags und ihrer Attribute hat man mit CSS sehr viele Möglichkeiten, Formulare zu gestalten. Es können alle allgemeinen, also nicht-spezifischen (wie z.B. list-style-type - außer natürlich es werden Listen im Formular verwendet) CSS-Eigenschaften angewandt werden.

Anzeigebeispiel - So sieht's aus: [formular.html](#)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">
<head>
<meta http-equiv="content-type" content="text/html; charset=UTF-8" />
<title>legend und fieldset</title>
<link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>

<form action="#" method="post">
<fieldset><legend>Absenderangaben</legend>
<label for="vorname" class="text">Vorname:</label>
```

```
<input type="text" name="vorname" id="vorname" size="20" maxlength="40"
/><br />
<label for="nachname" class="text">Nachname:</label>
<input type="text" name="nachname" id="nachname" size="20" maxlength="40" />
</fieldset>
</form>

</body>
</html>
```

```
body {
  font-family: Verdana, Helvetica, sans-serif;
  font-size: 0.8em;
}
```

```
fieldset {
  border: 2px solid green;
  background-color: lightgreen;
  line-height: 1.8em;
  width: 22em;
  -moz-border-radius: 5px;
  -webkit-border-radius: 5px;
  -khtml-border-radius: 5px;
  border-radius: 5px;
}
```

```
legend {
  border: 1px solid black;
  font-weight: bold;
  background-color: darkgreen;
  color: white;
  line-height: 1.4em;
  -moz-border-radius: 3.5px;
  -webkit-border-radius: 3.5px;
  -khtml-border-radius: 3.5px;
  border-radius: 3.5px;
}
```

```
label {
  float: left;
  text-align: right;
  width: 7em;
  margin-right: 1em;
  white-space: pre;
}
```

```
br {
  clear: left;
}
```

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_04:3_04_04_09



Last update: **2019/11/17 16:05**

3.4.4.10) Übungen zur Elementgestaltung

- [3.4.4.10.1\) Übung Schriften](#)
- [3.4.4.10.2\) Übung Boxmodell](#)
- [3.4.4.10.3\) Übung Formulare](#)
- [3.4.4.10.4\) Tabellen](#)
- [3.4.4.10.5\) Positionierung](#)
- [3.4.4.10.6\) Übung float und clear](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_04:3_04_04_10



Last update: **2019/11/17 16:11**

3.4.4.10.1) Übung Schriften

Gegeben ist das Gedicht: der Erbkönig.

- Formatiere die erste Strophe in sans-serif, die zweite Strophe in serif, die dritte Strophe in cursive, die vierte Strophe in fantasy, die fünfte Strophe in monospace.
- Verwende für jede einzelne Strophe eine Farbe.
- Nun verwende 4 verschiedene Eigenschaften(deiner Wahl) für das Gedicht.

Wer reitet so spät durch Nacht und Wind?
Es ist der Vater mit seinem Kind.
Er hat den Knaben wohl in dem Arm,
Er faßt ihn sicher, er hält ihn warm.

Mein Sohn, was birgst du so bang dein Gesicht?
Siehst Vater, du den Erbkönig nicht!
Den Erlenkönig mit Kron' und Schweif?
Mein Sohn, es ist ein Nebelstreif.

Du liebes Kind, komm geh' mit mir!
Gar schöne Spiele, spiel ich mit dir,
Manch bunte Blumen sind an dem Strand,
Meine Mutter hat manch gülden Gewand.

Mein Vater, mein Vater, und hörest du nicht,
Was Erlenkönig mir leise verspricht?
Sei ruhig, bleibe ruhig, mein Kind,
In dürren Blättern säuselt der Wind.

Willst finer Knabe du mit mir geh'n?
Meine Töchter sollen dich warten schön,
Meine Töchter führen den nächtlichen Reihn
Und wiegen und tanzen und singen dich ein

Wer reitet so spät durch Nacht und Wind?
Es ist der Vater mit seinem Kind.
Er hat den Knaben wohl in dem Arm,
Er faßt ihn sicher, er hält ihn warm.

Mein Sohn, was birgst du so bang dein Gesicht?
Siehst Vater, du den Erlkönig nicht!
Den Erlenkönig mit Kron' und Schweif?
Mein Sohn, es ist ein Nebelstreif.

Du liebes Kind, komm geh' mit mir!
Gar schöne Spiele, spiel ich mit dir,
Manch bunte Blumen sind an dem Strand,
Meine Mutter hat manch gülden Gewand.

Mein Vater, mein Vater, und hörest du nicht,
Was Erlenkönig mir leise verspricht?
Sei ruhig, bleibe ruhig, mein Kind,
In dürren Blättern säuselt der Wind.

Willst finer Knabe du mit mir geh'n?
Meine Töchter sollen dich warten schön,
Meine Töchter führen den nächtlichen Reihn
Und wiegen und tanzen und singen dich ein

Vorschau:

Lösung:

```
.absatz1{font-family:sans-serif; color:#00FF00; font-weight:bold;}  
.absatz2{font-family:serif; color:#0000FF; text-decoration:underline;}  
.absatz3{font-family:cursive; color:#00FFFF; text-decoration:overline;}  
.absatz4{font-family:fantasy; color:#008000;}  
.absatz5{font-family:monospace;}  
  
.text-shadow {text-shadow: #000000 3px 3px 5px;}
```

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_04:3_04_04_10:3_04_04_10_01

Last update: 2019/11/17 16:11



3.4.4.10.2) Übung Boxmodell



Erstelle eine Box mit den folgenden Eigenschaften:

- Länge: 1230px
- Höhe: 100px
- Hintergrundfarbe: hellblau
- Rahmen: Breite 3px / solide / rot gefärbt
- Textabstand zum Rahmen 20px
- Inhalt der Box:

„Gallia est omnis divisa in partes tres, quarum unam incolunt Belgae, aliam Aquitani, tertiam qui ipsorum lingua Celtae, nostra Galli appellantur. Hi omnes lingua, institutis, legibus inter se differunt. Gallos ab Aquitanis Garumna flumen, a Belgis Matrona et Sequana dividit.“

Hintergrundbild der Website:



From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_04:3_04_04_10:3_04_04_10_02



Last update: **2019/11/17 16:11**

3.4.4.10.3) Übung Formulare

Erstelle ein Formular mittels fieldset nach folgendem Muster:

(Bild folgt)

Folgende Formatierungen sollen umgesetzt werden:

- Formular:
 - Hintergrundfarbe: #eeeeff;
 - gepunkteter Rahmen: 1 Pixel, Abstand: 20 Pixel, in Grün
- fieldset:
 - 4 Pixel starker durchgezogener Rahmen in Silber

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_04:3_04_04_10:3_04_04_10_03



Last update: **2019/11/17 16:12**

3.4.4.10.4) Tabellenübung

Erstelle eine Tabelle, welche wie folgt aussehen soll:

ÜBUNGSTABELLE

1. Einheiten	Zeile 1	Eintrag 1
2. Einheiten	Zeile 2	Eintrag 2

Parameter

- Die Überschrift soll im <table>-Tag mit einer Caption erstellt werden.
- Die Schriftfarbe und das permanente Großschreiben (CSS-Tag: text-transform) in der Überschrift soll in einem externen Stylesheet erledigt werden.
- Die Einträge „1. Einheiten“ und „2. Einheiten“ sollen mittels table-header (th) erledigt werden. Die Formatierung dafür soll im Stylesheet passieren. (Schriftfarbe, Buchstabenabstand, ...)
- Backgrounds für die „Zeile x.“ und „Eintrag x.“ sollen im CSS-Stylesheet mittels Gruppen, die im HTML-Code einzufügen sind, definiert werden. (.bgblue und .bgred)
- Die Abstände der Einheiten vom Tabellenrand können mit dem CSS-Tag „padding: ??px“ eingestellt werden.

Mögliche Realisierung der Übung

Der HTML-Code ist hier einzusehen:

```
<html>

<head>
<title>&Uuml;bung - Tabellen</title>
<meta name="author" content="user">
<link rel="stylesheet" type="text/css" href="03_tabellen.css">
</head>

<body text="#000000" bgcolor="#FFFFFF" link="#FF0000" alink="#FF0000"
vlink="#FF0000">
<table border="1">
  <caption>&Uuml;bungstabelle</caption>
  <tr><th>1. Einheiten</th>
    <td class="bgblue">Zeile 1</td>
    <td class="bgred">Eintrag 1</td>
  </tr>
  <tr><th>2. Einheiten</th>
    <td class="bgblue">Zeile 2</td>
    <td class="bgred">Eintrag 2</td>
  </tr>
</table>
```

```
</tr>
</table>
</body>

</html>
```

Nützliche Links

[Tabellenformatierung mittels CSS in der Wiki erklärt](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_04:3_04_04_10:3_04_04_10_04



Last update: **2019/11/17 16:12**

3.4.4.10.5) Positionierungsübung

Erstelle zwei Boxen, welche sich folgendermaßen reihen:

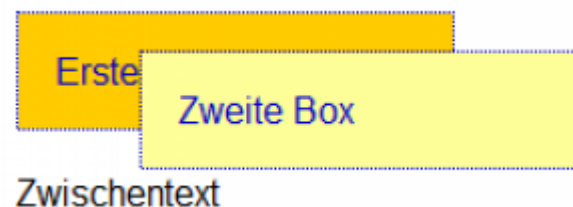


Zwischentext



(1.Version)

Danach ändere die Datei so, dass sich die Elemente so überlappen:



(2. Version)

Parameter

- Die Boxen sollen im CSS-Stylesheet für den HTML-Tag <div> definiert werden. (padding, border, width, color, ...)
- Für diese Boxen sollen zwei Klassen (.box1 und .box2) erstellt werden, in denen die Hintergrundfarbe und die Positionierung definiert werden.
- Die zweite Box soll bei der zweiten Version mittels „position:absolute;“ und den Einzügen „top: 2em;“ und „left: 4em;“ über die andere gelegt werden.

Mögliche Realisierung der Übung

Der HTML-Code mit CSS-Stylesheet ist hier einzusehen:

```
<html>

<head>
<title>Positionierungsübung</title>
<meta name="author" content="user">
<body text="#000000" bgcolor="#FFFFFF" link="#FF0000" alink="#FF0000"
vlink="#FF0000">
<style type="text/css">
body {
```



```
        font-family: Arial;
    }

div {
    width: 12em;
    color: #0000CC;
    border: dotted 1px;
    padding: 1em;
}

.box1 {
    background-color: #FFCC00;
    position:static;
}

.box2 {
    background-color: #FFFF99;
    position: absolute;
    left: 4em;
    top: 2em;
}
</style>
</head>

<body>
<div class="box1">Erste Box</div>
<br>Zwischentext<br><br>
<div class="box2">Zweite Box</div>
</body>
</html>
</body>

</html>
```

Nützliche Links

[Positionierung mittels CSS in der Wiki erklärt](#)

From:
<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_04:3_04_04_10:3_04_04_10_05

Last update: **2019/11/17 16:12**



3.4.4.10.6) Übung float und clear

Erstelle eine Seite mit einem Bild, das mithilfe von float und clear umflossen und dem durch overflow:hidden eine gewisse Breite zugewiesen wird.

Das Bild soll am linken Rand stehen und auf der rechten Seite umflossen werden.

So sähe es aus, wenn du den Code aus [Float- und Clear-Eigenschaft](#) kopiert hättest



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.



KATZENBILDER!!!! Such cute!



From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_04:3_04_04_10:3_04_04_10_06



Last update: **2019/11/17 16:12**

3.4.7) Übungen

- [3.4.5.1\) Übung 1 - Einbinden](#)
- [3.4.5.2\) Übung 2 - Einfache Elemente](#)
- [3.4.5.3\) Wiederholungsübung zu "XHTML-Elemente gestalten": Anmeldung bei XY](#)
- [3.4.5.4\) Screenshot einer Website nachbauen](#)
- [Abschließendes CSS-Quiz](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_05



Last update: **2019/11/17 16:16**

11.2.7.1) Übung 1 - Einbinden von CSS

- Erstellen eine CSS-Datei, die die folgenden Formate definiert:
 - h1: Schriftgröße 18pt, Farbe dunkelgrau
 - h4: Schriftgröße 14pt, Schriftgewicht normal
 - p: Schriftgröße 12pt, Farbe blau, Hintergrund hellgrau
- Validiere die CSS-Datei mithilfe des CSS-Validation Service des W3C
- Erstelle ein HTML-Dokument, das eine Überschrift 1. Ordnung und zwei Absätze, denen jeweils eine Überschrift 4. Ordnung vorausgeht, enthält. Binden Sie die externe CSS-Datei ein.
- Wandle das HTML-Dokument mithilfe von Tidy online in ein gültiges XHTML-Dokument um (sofern du nicht bereits ein XHTML-Dokument erstellt hast)
- Validiere das XHTML-Dokument.

Links

- [Try it Editor \(W3schools\)](#)

HTML Validierung

- [Markup Validation Service des W3C](#)
- [Web Design Group](#)
- [Validome](#)

CSS Validierung

- [CSSCheck der Web Design Group \(WDG\)](#)
- [CSS Validator des W3C](#)

HTML -> XHTML Konvertierung

- [Tidy Online](#)

From:
<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_05:3_04_05_01

Last update: **2019/11/17 16:15**



11.2.7.2) Übung 2 - Einfache Elemente

Erzeuge eine neue XHTML-Datei (quiz.html) und erstelle das Grundgerüst einer XHTML-Seite:

XHTML-Code

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">
  <head>
    <title> Quiz </title>
  </head>
  <body>
    <h2> Wer wird Auktionär </h2>
    <p>In diesem Quiz können Sie mit der richtigen Beantwortung von 15
Fragen
zum Auktionär werden. Als Hilfestellung stehen Ihnen die bekannten drei
Broker zur Verfügung:</p>
    <ul>
      <li>Publikumsbroker</li>
      <li>Anrufbroker</li>
      <li>Fifty/Fifty Broker</li>
    </ul>

    <p>Und hier kommt auch schon die erste Frage:</p>

    <div>Ein bekanntes Sprichwort lautet:
      <div>"Es ist noch kein Meister .."</div>
      <ol>
        <li>.. aus dem Bett gefallen.</li>
        <li>.. zur Ruhe gekommen.</li>
        <li>.. sitz' ich beim Schwager vorn.</li>
        <li>.. vom Himmel gefallen.</li>
      </ol>
    </div>
    <p>Herzlichen Glückwunsch, wollen Sie weitermachen ?</p>
  </body>
</html>
```

1. Lagere die CSS Angaben in eine externe Datei quiz.css aus und bette diese in dein HTML Dokument ein.
2. Modifiziere die Dokumenteigenschaften, indem du
 - die Hintergrundfarbe einstellst (background-color).
 - einen Rand links und rechts vorgibst (margin-left, margin-right).
 - die Schriftgröße um 20% vergrößerst. (font-size).
 - die Schriftart wechselst (font-family).
3. Verändere die Überschrift, indem du
 - die Schriftfarbe wechselst (color).

- den Text zentrierst (text-align).
 - die Schriftgrösse verdoppelst (font-size).
 - die Schriftart wechselst (font-family).
 - den Text unterstreichst (text-decoration).
4. Verändere die Quizfrage (<div> Element), indem du
 - einen Rahmen zeichnest (border).
 - die Hintergrundfarbe einstellst (background-color).
 - den Rahmen zentrierst (text-align).
 5. Verändere die Antwortliste (Element), indem du
 - Grossbuchstaben verwendest (list-style).
 - die Textfarbe wechselst.
 - den Text fett und kursiv setzt. (font-weight, font-style).
 6. Verändere die Brokerliste (Element), indem du
 - den runden Listenpunkt in einen quadratischen Listenpunkt änderst (list-style-type).
 - ein eigenes kleines Icon verwendest (im Internet [*.gif, *.png] suchen und runterladen) (list-style-image).
 7. Verändere die Dokumenteigenschaften, indem du
 - eine Hintergrundgrafik mit Farbverlauf verwendest.
 - Formatiere den Hintergrund so (background-attachment, etc.), wie es dir am besten gefällt.
 8. Experimentiere weiter, indem du
 - CSS-Elemente einbaust, die du noch nicht verwendet hast...

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_05:3_04_05_02



Last update: **2019/11/17 16:15**

11.2.7.3) Wiederholungsübung zu "XHTML-Elemente gestalten": Anmeldung bei XY

Der Benutzer soll sich bei „XY“ anmelden können.

- *Die folgenden Angaben sind nur die grundsätzlichen Richtlinien, selbstverständlich können weitere CSS-Eigenschaften und XHTML-Elemente verwendet werden!*
- Erstelle ein XHTML-Dokument und eine externes Stylesheet „style.css“
 - Die Schriftart für das gesamte Dokument soll Verdana sein
 - Der Hintergrund der gesamten Seite soll ein Farbverlauf von schwarz auf weiß sein (Stichwort: background-image und linear-gradient)
- Für alle Links gilt:
 - Sie sollen nicht unterstrichen sein
 - Sie sollen dunkelrot sein
 - Sie sollen bei Berührung mit der Maus schwarz werden
 - Während sie angeklickt (aber noch nicht losgelassen) werden, sollen sie fett sein
- Es soll eine DIV-Box geben, die
 - 65% des verfügbaren Platzes in der Breite nutzt, jedoch eine passende minimale Breite (die entsprechende Eigenschaft findet sich auf css4you) hat
 - die Hintergrundfarbe weiß hat
 - Einen beliebigen (nicht angegeben) Platz in die Höhe nutzt
 - Einen schwarzen Rahmen mit einer Strichstärke von 3px mit abgerundeten Ecken (Abrundungsgrad beliebig) hat
 - Einen Innenabstand von 7px hat
- In dieser DIV-Box soll
 - eine h1-Überschrift (die mit CSS passend formatiert wird, u.a. Schriftgröße 1.5em + schattiert) „Anmeldung bei XY“
 - eine h2-Überschrift (die mit CSS passend formatiert wird, u.a. Schriftgröße 1.2em) „Was ist XY?“
 - darunter ein Absatz sein, in dem den Nutzer in Fließtext darüber informiert wird, wozu er sich anmeldet. (Blindtext möglich!)
 - unter diesem Absatz eine h2-Überschrift „Vorteile bei XY“
 - zu dieser soll es eine Liste (Aufzählungszeichen Quadrat oder kleine Grafik) geben, in dem dem Nutzer Vorteile die er durch eine Mitgliedschaft hat aufgelistet werden
 - darunter soll erneut eine h2-Überschrift „Jetzt anmelden“ sein
 - zu dieser soll es ein ansprechend gestaltetes Anmeldeformular (mit Verwendung der neuen XHTML-Tags fieldset) geben, legend soll sein „Daten eingeben“
- Als Footer soll es eine weitere DIV-Box geben, die am unteren Rand der Seite positioniert wird.
 - Hintergrund dieser Box soll weiß sein
 - Außerdem soll es auf den oberen Seite der Box einen schwarzen Rahmen geben
 - darin soll es einen Copyright-Vermerk und einen Link `Impressum` geben
- Für den Fall das der der Benutzer die Seite als Anmeldeformular ausdrucken will, soll es ein entsprechendes Druck-Layout geben (medienspezifische CSS-Datei)
- Validiere die XHTML und die CSS Datei und beseitige alle Fehler
- Rechts neben der Haupt-DIV-Box soll es einen 120×480 Pixel großen vertikalen Banner geben. (Verwende den grünen Banner von unten, Tipp: float, weiteres allumfassendes DIV)



Banner

Mögliche Lösung

Anmeldung bei XY

Was ist XY?

XY ist die neue Lösung für die Kommunikation in Ihrer Firma. Wir bieten Ihnen ein schnelles, sicheres und zuverlässiges Sprach- und Textkommunikationssystem, welches über mehrere Netzwerke seine Vorteile behält.

Vorteile bei XY

- Jederzeit kündbar
- Schnell
- Sicher
- Zuverlässig
- Einzigartige klare Sprachqualität
- Keine Authentifizierungsprobleme dank Andbindung der Clients an die Hardware

Jetzt anmelden

Daten eingeben

Username:

E-Mail:

Passwort:

Passwort wiederholen:

Banner

Copyright: Impressum

HTML-Code

```
<html>

<head>
<title>Anmeldung bei XY</title>
<meta name="author" content="">
</head>

<body text="#000000" bgcolor="#FFFFFF" link="#FF0000" alink="#FF0000"
vlink="#FF0000">

<div>

<h1>Anmeldung bei XY</h1>
<h2>Was ist XY?</h2>
XY ist die neue Lösung für die Kommunikation in Ihrer Firma.<br>
Wir bieten Ihnen ein schnelles, sicheres und zuverlässiges Sprach- und
Textkommunikationssystem,<br>
```

```
welches &uuml;ber mehrere Netzwerke seine Vorteile beh&auml;lt.<br>
<h2>Vorteile bei XY</h2>
<ul>
  <li>Jederzeit k&uuml;ndbar</li>
  <li>Schnell</li>
  <li>Sicher</li>
  <li>Zuverl&auml;ssig</li>
  <li>Einzigartige klare Sprachqualit&auml;t</li>
  <li>Keine Authentifizierungsprobleme dank Andbindung der Clients an die
Hardware</li>
</ul>
<h2>Jetzt anmelden</h2>

<form action="anmeldung.html" method="POST">
<fieldset><legend>Daten eingeben</legend>
Username: <input type="Text" name="username" value="Username" size="40"
maxlength="40"><br>
E-Mail: <input type="Text" name="email" value="E-Mail" size="50"
maxlength="50"><br>
Passwort: <input type="Password" name="passwort" size="30"
maxlength="30"><br>
Passwort wiederholen: <input type="Password" name="passwort1" size="30"
maxlength="30"><br>
<input type="Submit" name="absenden" value="Absenden"> <input
type="reset"><br>
</fieldset>
</form>
</div>

<div>
  Copyright: <a href=„#">Impressum</a>
</div>

</body>

</html>
```

From:
<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_05:3_04_05_03



Last update: 2019/11/17 16:15

11.2.7.4) Übung: Screenshot einer Website nachbauen

Baue folgende Webseite möglichst originalgetreu nach.

Hinweise:

- der hervorgehobene Menüpkt wird gerade mit der Maus berührt
- die Gesamtbreite soll immer 70% des verfügbaren Platzes ausmachen, der Menübereich soll immer gleich breit bleiben (128px)
- das „Logo“ rechts oben soll eine Bilddatei sein!



From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_05:3_04_05_04

Last update: 2019/11/17 16:15



3.4.6) Links

- [CSS4You](#): deutschsprachige CSS-Referenz mit vielen Informationen
- [Übersicht über Neuerungen in CSS3](#) (webmasterpro.de)
- [CSS auf 4webmaster.de](#)
- [Webseiten-Beispiele von Sebastian](#)
- [CSS-Workshop](#)
- [W3Schools](#): englischsprachige Referenz-, Übungs- und Lernplattform
- [Buch: Jetzt lerne ich CSS](#)
- [Validator \(W3\)](#):
 - [CSSa](#)
 - [Unicorn \(HTML + CSS + Feed\)](#)
- [browsershots.org](#)
- [CSS HOW-TOs, CSS3-Features und Templates](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_06



Last update: **2019/11/17 16:17**

NETZWERKE

Allgemeine Netzwerkgrundlagen

- [Grundlagen](#)
- [Topologien](#)
- [Übertragungsmedien](#)
- [Ethernet + Zugriffsverfahren CSMA/CD](#)
- [Schichtenmodell](#)
- [Netzwerkgeräte](#)
- [Adressierung](#)
 - [Adressierung Übungen](#)
- [Netzwerksimulation mit FILIUS](#)
- [Routing](#)
- [Netzwerkbefehle](#)
- [Protokolle](#)
- [Netzwerkanalyse mit Wireshark](#)
 - [Wireshark Übungen](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4

Last update: **2020/03/17 21:32**



4.1) Netzwerk-Grundlagen

Was ist ein Netzwerk?

Ein Netzwerk ist die **physikalische und logische Verbindung von Computersystemen**. Ein einfaches Netzwerk besteht aus zwei Computersystemen. Sie sind über ein Kabel miteinander verbunden und somit in der Lage ihre Ressourcen gemeinsam zu nutzen. Wie zum Beispiel Rechenleistung, Speicher, Programme, Daten, Drucker und andere Peripherie-Geräte. Ein netzwerkfähiges Betriebssystem stellt den Benutzern auf der Anwendungsebene diese Ressourcen zur Verfügung.

Notwendigkeit für ein Netzwerk

Als es die ersten Computer gab, waren diese sehr teuer. Peripherie-Geräte und Speicher waren fast unbezahlbar. Zudem war es erforderlich zwischen mehreren Computern Daten auszutauschen. Aus diesen Gründen wurden Computer miteinander verbunden bzw. vernetzt.

Daraus ergaben sich einige Vorteile gegenüber unvernetzten Computern:

- zentrale Steuerung von Programmen und Daten
- Nutzung gemeinsamer Datenbeständen
- erhöhter Datenschutz und Datensicherheit
- größere Leistungsfähigkeit
- gemeinsame Nutzung der Ressourcen

Die erste Möglichkeit, Peripherie-Geräte gemeinsam zu nutzen, waren manuelle Umschaltboxen. So konnte man von mehreren Computern aus einen Drucker nutzen. An welchem Computer der Drucker angeschlossen war, wurde über die Umschaltbox bestimmt. Leider haben Umschaltboxen den Nachteil, dass Computer und Peripherie beieinander stehen müssen, weil die Kabellänge begrenzt ist.

Größenordnung von Netzwerken

Jedes Netzwerk basiert auf Übertragungstechniken, Protokollen und Systemen, die eine Kommunikation zwischen den Netzwerk-Teilnehmern ermöglichen. Bestimmte Netzwerktechniken unterliegen dabei Beschränkungen, die insbesondere deren Reichweite und Ausdehnung begrenzt. Hierbei haben sich verschiedene Netzwerk-Dimensionen durchgesetzt für die es unterschiedliche Netzwerktechniken gibt.

- PAN - Personal Area Network: personenbezogenes Netz, z. B. Bluetooth
- LAN - Local Area Network: lokales Netz, z. B. Ethernet
- MAN - Metropolitan Area Network: regionales Netz
- WAN - Wide Area Network: öffentliches Netz, z. B. ISDN
- GAN - Global Area Network: globales Netz, z. B. das Internet

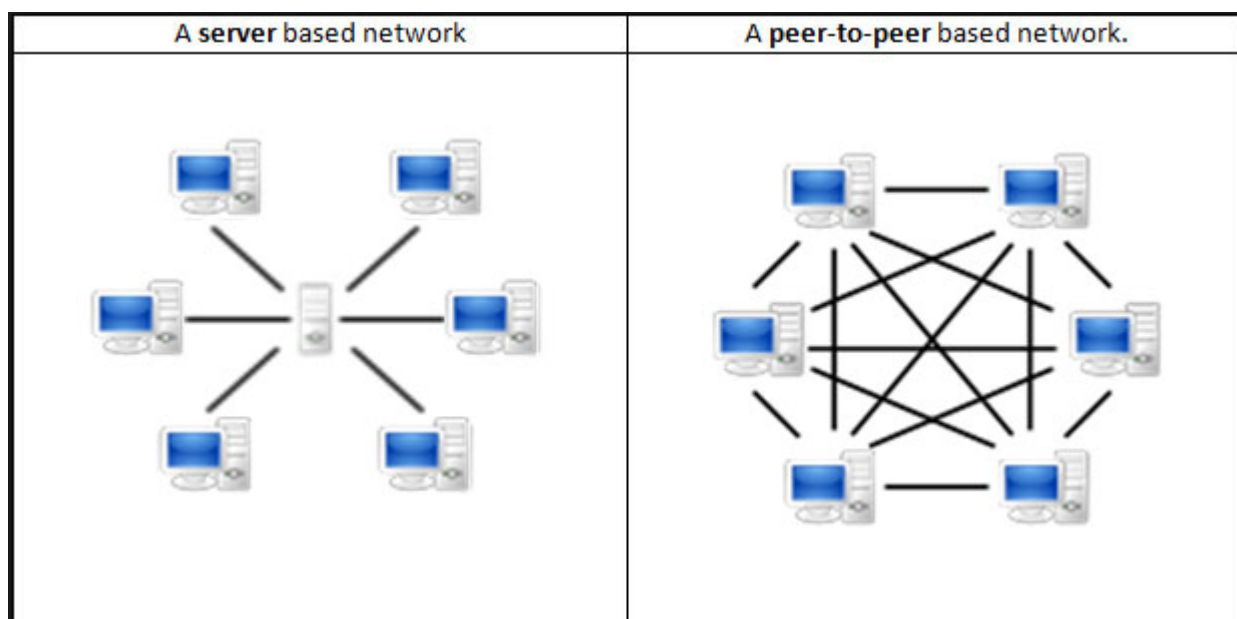
Einteilung nach Ausdehnung



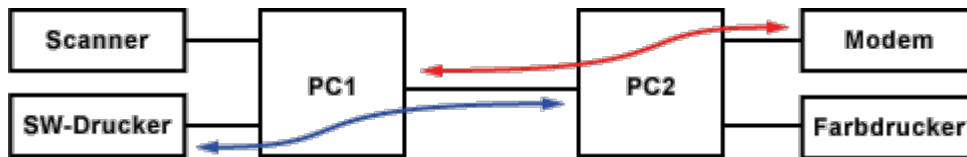
In der Regel findet ein Austausch zwischen den Netzen statt. Das heißt, dass Netzwerk-Teilnehmer eines LANs auch ein Teilnehmer eines WANs oder eines GANs ist. Eine 100%ig klare Abgrenzung zwischen diesen Dimensionen ist nicht immer möglich, weshalb man meist nur eine grobe Einteilung vornimmt. So unterscheidet man in der Regel zwischen LAN und WAN, wobei es auch Techniken und Protokolle gibt, die sowohl im LAN, als auch im WAN zum Einsatz kommen.

Peer-to-Peer-Netze und Client-Server-Architekturen

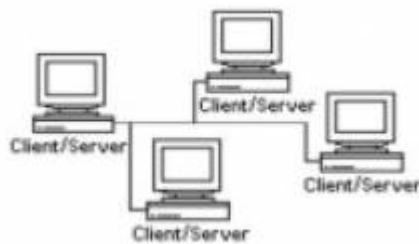
Man unterscheidet zwei „Philosophien“:



Peer-to-Peer-Netzwerke



In einem Peer-to-Peer-Netzwerk ist jeder angeschlossene Computer zu den anderen gleichberechtigt. Jeder Computer stellt den anderen Computern seine Ressourcen zur Verfügung. Ein Peer-to-Peer-Netzwerk eignet sich für bis zu 10 Stationen. Bei mehr Stationen wird es schnell unübersichtlich. Diese Art von Netzwerk ist relativ schnell und kostengünstig aufgebaut. Die Teilnehmer sollten möglichst dicht beieinander stehen. Einen Netzwerk-Verwalter gibt es nicht. Jeder Netzwerk-Teilnehmer ist für seinen Computer selber verantwortlich. Deshalb muss jeder Netzwerk-Teilnehmer selber bestimmen, welche Ressourcen er freigeben will. Auch die Datensicherung muss von jedem Netzwerk-Teilnehmer selber vorgenommen werden.

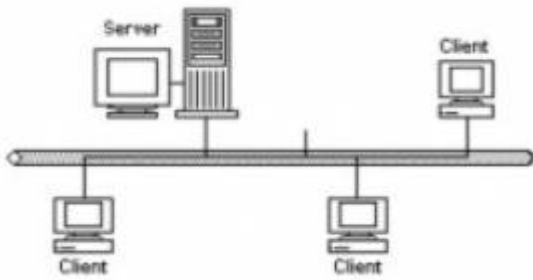


Peer-to-Peer-Netze brauchen keinen eigenen Server-Rechner, da jeder PC Server-Funktionen übernehmen kann.

Client/Server-Architekturen



In einem serverbasierten Netzwerk werden die Daten auf einem zentralen Computer gespeichert und verwaltet. Man spricht von einem dedizierten Server, auf dem keine Anwendungsprogramme ausgeführt werden, sondern nur eine Server-Software und Dienste ausgeführt werden. Diese Architektur unterscheidet zwischen der Anwender- bzw. Benutzerseite und der Anbieter- bzw. Dienstleisterseite. Der Anwender betreibt auf seinem Computer Anwendungsprogramme (Client), die die Ressourcen des Servers auf der Anbieterseite zugreifen. Hier werden die Ressourcen zentral verwaltet, aufgeteilt und zur Verfügung gestellt. Für den Zugriff auf den Server (Anfrage/Antwort) ist ein Protokoll verantwortlich, dass sich eine geregelte Abfolge der Kommunikation zwischen Client und Server kümmert. Die Client-Server-Architektur ist die Basis für viele Internet-Protokolle, wie HTTP für das World Wide Web oder SMTP/POP3 für E-Mail. Der Client stellt eine Anfrage. Der Server wertet die Anfrage aus und liefert eine Antwort bzw. die Daten zurück.



From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4:4_00

Last update: **2020/01/19 15:47**



4.2) Netzwerk-Topologien

Die Struktur eines Netzwerks bezeichnet man als Topologie. Wie wichtig die Struktur eines Netzwerks ist, merkt man bei einem Leitungsausfall: ein gutes Netzwerk findet bei einem Leitungsausfall selbstständig einen neuen Pfad zum Empfänger.

Physikalische und logische Topologie

Interessant ist, dass sich die **sichtbare Topologie** (also die physische Verkabelungsstruktur) vom tatsächlichen Datenfluss unterscheiden kann. Deshalb verwendet man für die hardwaremäßige Realisierung den Begriff **physikalische Topologie** während man für den tatsächlichen Datenfluss den Begriff „logische Topologie“ verwendet.

Die wichtigsten Netzwerktopologien sind:

Bus-Topologie



ALLE GERÄTE nutzen DASSELBE KABEL

Bei einem Bussystem sind alle Rechner hintereinander geschaltet und über Abzweige (T-Stücke) an das Netzkabel angeschlossen. Problem: Eine Verbindungsunterbrechung betrifft den ganzen Bus!

Vorteile

- Relativ niedrige Kosten, da geringe Kabelkosten
- Ausfall einer Station führt zu keinem Netzausfall

Nachteile

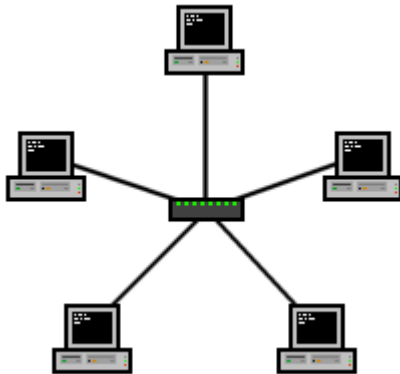
- Alle Daten über ein Kabel
- Nur eine Station kann senden. Alle anderen sind blockiert.
- Eine Störung an einer Stelle (z.B.: Defektes Kabel) führt zu einem Netzausfall (⇒ aufwendige Fehlersuche)
- Unverschlüsselter Netzwerkverkehr kann direkt am Bus (=Kabel) mitgelesen werden

Einsatzgebiet

Früher aufgrund der niedrigen Kosten häufig verwendet, heute spielt die Bus-Topologie keine Rolle mehr und wurde von der Stern-Topologie verdrängt.

[W Bus-Topologie - Details](#)

Stern-Topologie



JEDES GERÄT nutzt EIN KABEL.

Damit ist es zu einem Verteiler verbunden. Es existiert eine Punkt-zu-Punkt Verbindung zwischen Verteiler und Gerät. Als Verteiler kann ein HUB oder ein SWITCH dienen.

Vorteile

- Ausfall einer Station oder eines Defekts an einem Kabel führt zu keinem Netzausfall
- Aktive Verteiler (Switch, Hub) dienen gleichzeitig als Signalverstärker
- Bei richtiger Konfiguration können zwei Stationen die volle Bandbreite des Übertragungsmediums für ihre Kommunikation nutzen, ohne andere Stationen dabei zu behindern. Diese physikalische Topologie erlaubt somit sehr hohe Datendurchsatzraten.
- Weitere Stationen oder Verteiler können einfach hinzugefügt werden. Sehr leicht skalierbar.

Nachteile

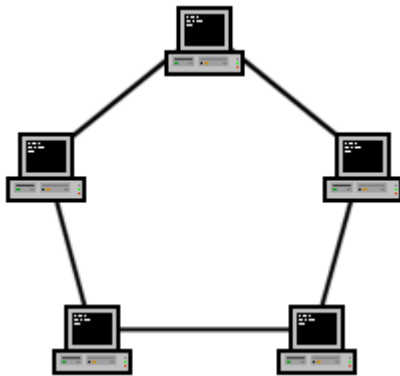
- Große Kabelmengen
- Beim Ausfall des Verteilers ist kein Netzverkehr mehr möglich

Einsatzgebiet

Im praktischen Einsatz bei lokalen Netzwerken findet die Stern-Topologie Verwendung. Häufigste Form der Verkabelung.

[W Stern-Topologie - Details](#)

Ring-Topologie



JEDES GERÄT ist mit ZWEI NACHBARN verbunden.

Die Ring-Topologie ist eine geschlossene Form, es gibt keinen Kabelanfang und kein Kabelende. Es handelt sich jeweils um eine Punkt-zu-Punkt Verbindung zwischen den Rechnern. Jede Station hat genau einen Vorgänger und einen Nachfolger. Datenverkehr findet immer nur in eine Richtung statt.

Vorteile

- Vorgänger und Nachfolger sind festgelegt
- Alle Stationen verstärken das Signal
- Alle Stationen haben gleiche Zugriffsmöglichkeit
- Leicht umsetzbar

Nachteile

- Ausfall einer Station oder eines Kabelteils führt zu einem Netzausfall
- Hoher Aufwand bei der Verkabelung (Jede Station braucht 2 Netzwerkkarten)
- Leicht abhörbar
- langsame Datenübertragung bei vielen Stationen

Einsatzgebiet

Physikalische Anwendung gibt es heute keine mehr.
Logische Anwendung findet sie im Token Ring.

[W Ring-Topologie - Details](#)

Mischformen

Sind zumeist **Kombinationen aus Bus, Stern und Ring.**

Backbone

Unter einem Backbone („Rückgrat“) wird die physikalische Verbindung zwischen einzelner Teilnetze verstanden. Es wird auch oft als Hintergrundnetz betitelt und verbindet z.B. mehrere Gebäude.

Stern-Bus-Netz

Ein Stern-Bus-Netz entsteht, wenn mehrere Verteiler über einen Bus miteinander verbunden sind. Häufig sind so mehrere Stockwerke miteinander verbunden.



Stern-Stern-Netz

Ein Stern-Stern-Netz entsteht, wenn mehrere Verteiler wiederum über einen Verteiler verbunden sind. Häufigste Anwendung ist wiederum das Verbinden von mehreren Subnetzen (z.B. Netze in verschiedenen Stockwerken).

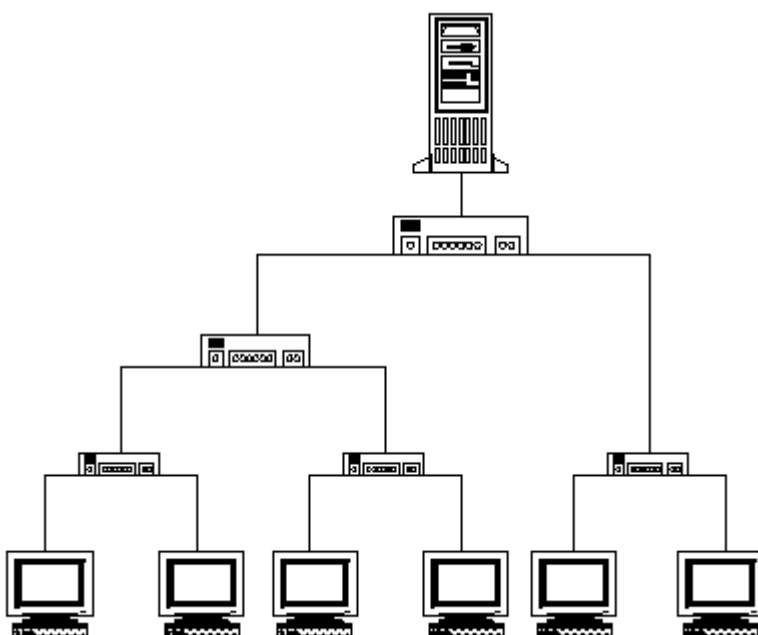
Fällt der Hauptverteiler aus, so kann zwischen den Stockwerken nicht mehr kommuniziert werden. Jedoch kann man auch die Hauptverteiler redundant auslegen.



Baum

Eine Baum-Topologie wird so aufgebaut, dass, ausgehend von der Wurzel, eine Menge von Verzweigungen zu weiteren Verteilungsstellen existiert.

Es handelt sich somit um eine Erweiterung der Stern-Stern-Topologie auf mehrere Ebenen.



Maschen-Topologie



Vorherrschende Netzstruktur in großflächigen Netzen (z.B. öffentliche Telekommunikationsnetze).

W [Maschen-Topologie - Details](#)

Zell-Topologie

Die Zell-Topologie kommt hauptsächlich bei drahtlosen Netzen zum Einsatz. Eine Zelle ist der Bereich um eine Basisstation (z.B. Wireless Access Point), in dem eine Kommunikation zwischen den Endgeräten und der Basisstation möglich ist.

W [Zell-Topologie - Details](#)

Zusammenfassung

Topologie	Vorteile	Nachteile
Bus-Topologie	<ul style="list-style-type: none"> • einfach installierbar • kurze Leitungen 	<ul style="list-style-type: none"> • Netzausdehnung begrenzt • bei Kabelbruch fällt Netz aus • aufwändige Zugriffsmethoden
Ring-Topologie	<ul style="list-style-type: none"> • verteilte Steuerung • große Netzausdehnung 	<ul style="list-style-type: none"> • aufwendige Fehlersuche • bei Störungen Netzausfall • hoher Verkabelungsaufwand
Stern-Topologie	<ul style="list-style-type: none"> • einfache Vernetzung • einfache Erweiterung • hohe Ausfallsicherheit 	<ul style="list-style-type: none"> • hoher Verkabelungsaufwand • Netzausfall bei Ausfall oder Überlastung des Hubs
Maschen-Topologie	<ul style="list-style-type: none"> • dezentrale Steuerung • unendliche Netzausdehnung • hohe Ausfallsicherheit 	<ul style="list-style-type: none"> • aufwendige Administration • teure und hochwertige Vernetzung



Netztopologien

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4:4_01

Last update: **2020/01/19 15:48**



4.3) Übertragungsmedien

Die Übertragungsmedien sind die Straßen der Daten. Der Aufbau dieser Straßen muss sehr gut geplant werden, um alle aktuellen Anforderungen bzw. eventuelle zukünftige Anforderungen ohne großartige Veränderungen zu erfüllen.

Als **Maßeinheit für die Übertragungsgeschwindigkeit** werden die Werte in **bit/s, b/s bps, -> also Bit pro Sekunde** angegeben. Achtung: Nicht zu verwechseln mit **Byte/s -> Byte pro Sekunde!!**

$$C = D/t \quad \text{` (bits)/(s) `}$$

1) Rechenbeispiel:

Es werden 100MB in 10s übertragen. Wie hoch ist die Übertragungsgeschwindigkeit?

- $D=100\text{MByte}$
- $t=10\text{s}$

Rechenschritt	Berechnung
D umwandeln in bits	$100 \cdot 1024 \cdot 1024 \cdot 8 = 838860800 \quad \text{` bits `}$
C berechnen	$C = D/t = (838\,860\,800)/10 = 83886080 \quad \text{` (bit)/(s) `}$
C umwandeln in Mbit/s	$(83886080)/(1024)/(1024) = 80 \quad \text{` (Mbit)/(s) `}$

2) Rechenbeispiel:

Max hat eine Datentransferrate von 10Mbit/s Download und 2Mbit/s Upload.

a) Wie lange braucht er, um 10MB runterzuladen?

- $D=10\text{MB}$
- $C=10\text{Mbit/s}$

Rechenschritt	Berechnung
D umwandeln in bits	$10 \cdot 1024 \cdot 1024 \cdot 8 = 83886080 \quad \text{` bits `}$
C umwandeln in bit/s	$10 \cdot 1024 \cdot 1024 = 10485760 \quad \text{` (bit)/(s) `}$
Formel umformeln	$t = (D)/(C) \quad \text{` (bits)/((bit)/s) `}$
In Formel einsetzen	$t = 83886080/10485760 = 8 \quad \text{` s `}$

b) Wie lange braucht er, um 10MB hochzuladen?

Rechenschritt	Berechnung
D umwandeln in bits	$10 \cdot 1024 \cdot 1024 \cdot 8 = 83886080 \quad \text{` bits `}$
C umwandeln in bit/s	$2 \cdot 1024 \cdot 1024 = 2097152 \quad \text{` (bit)/(s) `}$
Formel umformeln	$t = (D)/(C) \quad \text{` (bits)/((bit)/s) `}$
In Formel einsetzen	$t = 83886080/2097152 = 40 \quad \text{` s `}$

Leitergebundene Übertragung

Bei einer leitergebundenen Übertragung werden Medien in Form von Kabeln benötigt (Metallische Leiter, Glasfaser).

Ein Kabel besteht zumindest aus einer Ader (=Faser).

Mehrere Adern sind durch entsprechende Isolationsschichten getrennt.

Alle Adern wiederum werden von einem Mantel als Schutz umgeben.

Die Übertragung selbst erfolgt durch elektromagnetische Schwingungen.

Koaxialkabel

Das früher verwendete Koaxialkabel ist in modernen Netzen praktisch vollständig von Twisted Pair-Kabeln (TP) und Lichtwellenleiter (LWL) abgelöst worden.



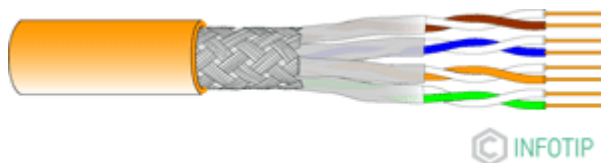
Es besteht aus

- einem Innenleiter (Kupfer, Stahlkupfer)
- einer Isolation (Dielektrikum)
- einer Abschirmung (Metallgeflecht schützt vor magnetischen Störungen -> Rauschen & Übersprechen)
- einem Mantel

Es waren bis zu 10Mbps möglich:

- Thicknet (10Base5)
- Thinnet (10Base2) - Heute noch bei Satellitenempfang im Einsatz


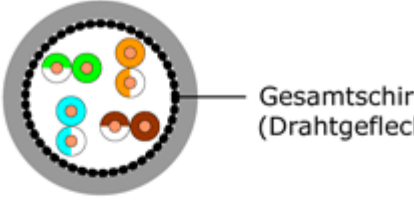
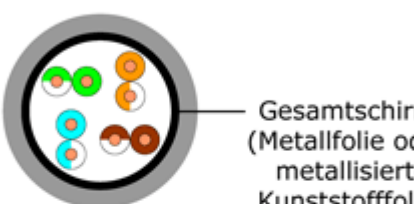
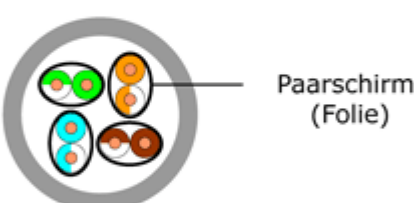
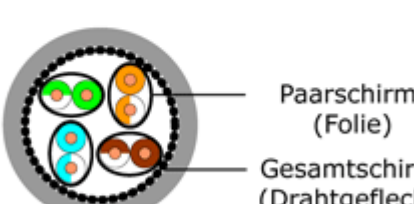
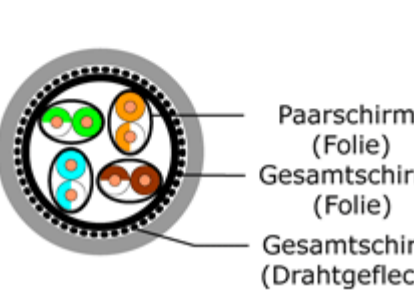
Twisted-Pair Kabel



Twisted Pair ist ein vier-, acht- oder mehr-adriges Kupferkabel, bei dem jeweils zwei Adern miteinander verdreht sind. Durch die Verdrehung kompensieren sich Leitungskapazität und -induktivität. Dadurch steigt die Übertragungsbandbreite und die mögliche Übertragungsreichweite wird praktisch nur durch die Dämpfung des Wirkwiderstandes begrenzt. Die Verwendung von symmetrischen Signalen (Differentialspannungen) erhöht die Festigkeit gegen elektromagnetische Störstrahlung.



Twisted Pair-Kabel gibt es in zahlreichen Varianten. Twisted Pair-Verbindungen werden außer in der Kommunikationstechnik (Netzwerkkabel, Telefonkabel) auch bei HDMI-, DVI- und LVDS-(in LCD/Plasma-TV zwischen Signalprozessor und Display) Verbindungen eingesetzt. Die Anzahl der Leiterpaare im Kabel hängt dabei von der benötigten Datenübertragungsrate ab. In Netzwerken wird für jede Übertragungsrichtung (senden, empfangen) wird jeweils ein Adernpaar (bei 100BaseT4 und 1000BaseT jeweils zwei) genutzt. Die Übertragungsreichweite ist abhängig vom Aufbau des Kabels, von der Dämpfung (=Länge) des Kabels und von den externen Störeinflüssen. Twisted Pair-Kabel für Netzwerke gibt es in zahlreichen Varianten:

Bild	Benennung	Beschreibung
U/UTP (UTP) 	U/UTP-Kabel	Unshielded/Unshielded Twisted Pair sind nicht abgeschirmte verdrehte Leitungen und gehörten früher typischerweise der CAT3 an. UTP-Kabel sollten im industriellen Bereich oder in der Datentechnik mit hohen Datenraten nicht verwendet werden.
S/UTP (S/UTP) 	S/UTP-Kabel	Screened/Unshielded Twisted Pair haben einen Gesamtschirm aus einem Kupfergeflecht zur Reduktion der äußeren Störeinflüsse
F/UTP (F/UTP) 	F/UTP-Kabel	Foilshielded/Unshielded Twisted Pair besitzen zur Abschirmung einen Gesamtschirm, zumeist aus einer alukaschierten Kunststofffolie
U/FTP (FTP) 	U/FTP-Kabel	Unshielded/Foilshielded Twisted Pair auch genannt CAT5 oder CAT5e . Die Leitungsadern sind paarweise mit Folie abgeschirmt
S/FTP (S/FTP) 	S/FTP-Kabel	Screened/ Foilshielded Twisted Pair auch genannt CAT6 sollten in Bereichen mit hoher Störstrahlung (z.B. Büros mit mehreren PCs) eingesetzt werden.
SF/FTP (SF/FTP) 	SF/FTP-Kabel	Screened Foilshielded/Foilshielded Twisted Pair auch genannt CAT6e oder CAT7 besitzen eine Abschirmung für jedes Kabelpaar sowie eine doppelte Gesamtschirmung. Hierdurch kann eine optimale Störleistungsunterdrückung erreicht werden. Auch das Übersprechen zwischen den einzelnen Adernpaaren wird so wirksam unterdrückt

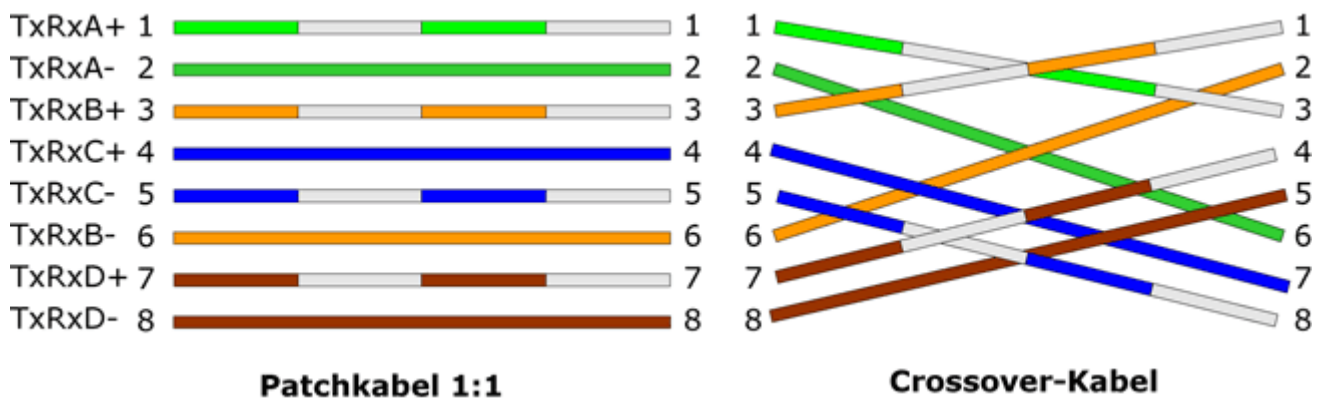
Die Preisunterschiede zwischen CAT-5e- Kabeln und CAT-7-Kabeln ist so gering, dass es sich bei Neuinstallation auf jeden Fall empfiehlt, CAT-7-Kabel einzusetzen. Dieses ist als einziges Kupfermedium in der Lage mit dem kommenden 10Gbit-LAN verwendet zu werden.

Verbinder - RJ45



Der typische Standardverbinder für die Twisted-Pair-Verkabelung eines kupfergebundenen Ethernet-Netzwerkes ist der **8polige Western-Modularstecker RJ-45** (8P8C), auch RJ-48 oder RJ-49 genannt. RJ-45 Steckverbindungen können auf zwei Arten belegt sein, wobei die Belegung nach T568B am weitesten verbreitet zu sein scheint:

Belegung nach EIA/T-T568A		Belegung nach EIA/T-T568B	
Pin	Farbe	Pin	Farbe
1	weiß-grün	1	weiß-orange
2	grün	2	orange
3	weiß-orange	3	weiß-grün
4	blau	4	blau
5	weiß-blau	5	weiß-blau
6	orange	6	grün
7	weiß-braun	7	weiß-braun
8	braun	8	braun



Bei 1:1-Verbindungen sind beide Beschaltungen elektrisch zueinander kompatibel. Nur bei Erweiterungen von fest verdrahteten Netzen ist festzustellen, welche Belegung bereits vorgegeben ist. Normale Verbindungskabel („Patchkabel“) mit RJ-45-Steckern sind 1:1 verschaltet, d.h. Pin 1 des einen Steckers geht auf den Pin 1 des anderen Steckers usw. Nur in besonderen Fällen, wenn z.B.

zwei Netzwerkkarten direkt miteinander verbunden werden sollen oder wenn Netzwerkkomponenten (z.B. Hubs älterer Bauart) über keinen dedizierten Uplink-Port verfügen, kann der Einsatz von Crossover-Kabeln notwendig werden.



LichtWellenLeiter (LWL)

Sind mit der Netzwerkverkabelung weite Strecken zu überwinden, z.B. zwischen einzelnen Gebäuden auf einem Fabrikgelände („Campusbereich“), sind sehr hohe Datenübertragungsraten (z.Zt. bis zu 170Gb/s) gefordert oder wenn sich die Datenübertragung per Kupferkabel aus technischen Gründen (z.B. bei extremer Störstrahlung) oder aus Gründen der Sicherheit verbietet, werden Lichtwellenleiter (LWL, Glasfasern) als Übertragungsmedium eingesetzt. Die Lieferprogramme der Hersteller erlauben mittlerweile die Übertragungsstrecken bis zum Einzelplatz komplett auf der Basis von LWL auszuführen.



Aufbau und Prinzip

In einem LWL werden die Informationen nicht, wie in einem Kupferkabel, elektrisch übertragen, sondern mit **Licht**.

Der eigentliche LWL ist eine Faser aus Glas oder Kunststoff. Jede Faser besteht aus zwei Schichten. Der konzentrische Kern besteht aus einem optischen Material mit einem hohen Brechungsindex, das Mantelglas („Cladding“) aus einem Material mit niedrigem. Licht, das in einem bestimmten Winkelbereich auf den Übergang von Kern zum Mantel trifft wird dort vollständig reflektiert. Über

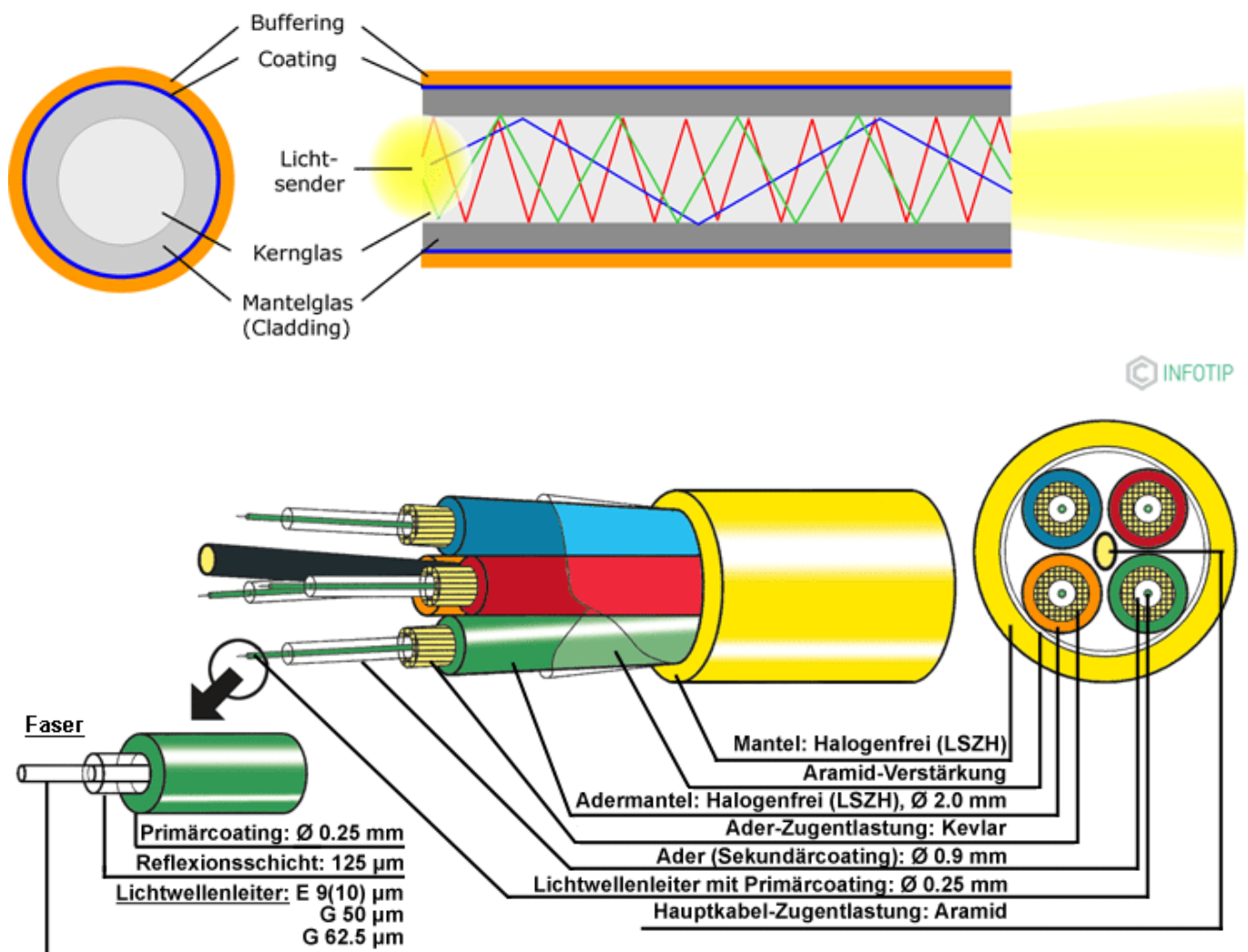
solche fortlaufenden Totalreflexionen pflanzt sich das Licht durch den LWL bis zum Ende der Faser fort.

Je steiler der Einfallswinkel des Lichts bei der Einspeisung in den LWL ist, desto häufiger wird die Lichtwelle reflektiert. Mit jeder Reflektion der Lichtwelle wird der Weg, des sogenannten Modes, länger.

Licht, das wenig häufig reflektiert wird, hat einen kürzeren Weg und durchläuft die Faser schneller. Es ist Licht niedrigen Modes.

Licht, das sehr häufig reflektiert wird, hat eine niedrige Ausbreitungsgeschwindigkeit in der Faser. Er ist Licht hohen Modes.

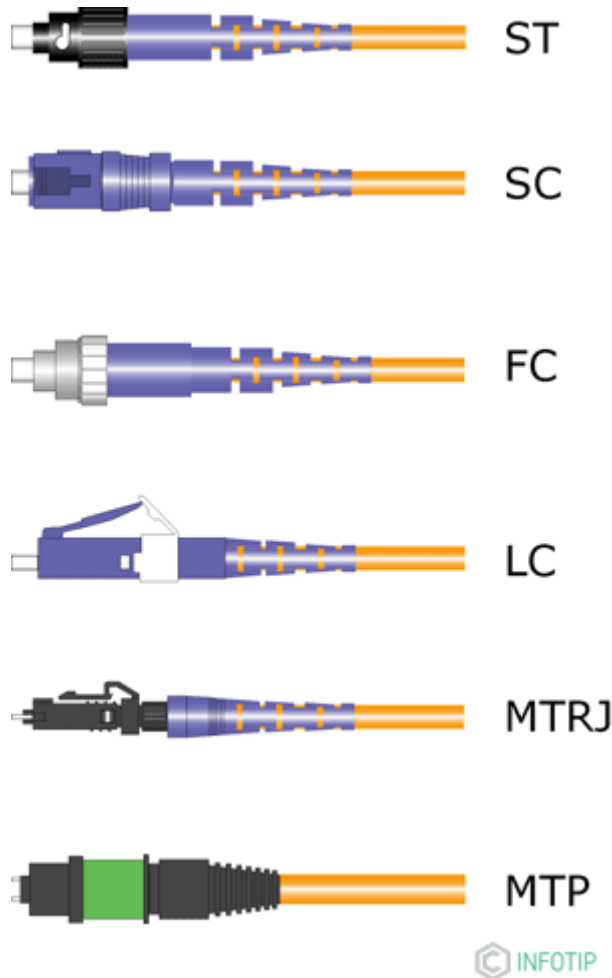
Erzeugt die Lichtquelle des Senders ein nicht-kohärentes Licht, tritt das Licht mit einer Vielzahl unterschiedlicher Winkel in die Faser ein. Dadurch entstehen natürlich durch die unterschiedlichen Moden Laufzeitunterschiede zwischen den Signalanteilen. Ein Eingangsimpuls mit steilen Flanken wird dadurch verschliffen und in seiner Breite gedehnt. Je länger ein Kabel ist, desto höher wird auch diese sog. Dispersion (Einheit: ns/km). Die Dispersion beeinflusst direkt die Übertragungsbandbreite der Glasfaserverbindung.



Da die Fasern sehr dünn und empfindlich sind, werden sie zum mechanischen Schutz mit einer Kunststoffbeschichtung („Coating“) und einem Schutzüberzug versehen. In einem LWL-Kabel können mehrere Fasern, sogar in mehreren Bündeln, zusammen gefasst sein.

Verbinder

Die Hersteller von Netzwerkzubehör bieten konfektionierte Verbindungs- und Patchkabel mit einer Vielzahl von verschiedenen Steckerformen an. Meist sind die Kabel paarweise angelegt um beide Datenflussrichtungen (TX und RX) gleichzeitig herstellen zu können.



Vorteile

- hohe Reichweite
- hohe Übertragungsbandbreite
- Potentialfrei, daher auch für explosionsgefährdete Bereiche geeignet
- hohe Störfestigkeit, LWL können sogar zu Energieversorgungskabeln parallel verlegt werden
- hohe Abhörsicherheit

Nachteile

- Material für die Verkabelung ist teuer
- teure Verbindungstechnik
- Die Montagekosten sind wegen des höheren technischen Aufwandes höher
- komplexe und teure Messtechnik
- zusätzliche Kosten für Medienkonverter auf Kupfer-Ethernet

Leiterungebundene Übertragung

Als leiterungebundene Übertragung bezeichnet man eine Übertragung per

- Funk
- Ultraschall
- Infrarot
- Laser
- Licht

Drahtlose Übertragung (WLAN)

Die Übertragung von Informationen ohne Kabel ist mittlerweile in vielen Lebensbereichen als praktische Alternative eingezogen. Von der Fernbedienung eines Fernsehers über drahtlose Lautsprecher bis zum Smartphone gibt es viele Beispiele für die Umsetzung dieser Technik. Dabei werden Funksignale in frei verfügbaren Frequenzbändern anstelle von Kabeln für die Datenübertragung verwendet.

Vorteile

- Es sind keine baulichen Maßnahmen innerhalb eines Gebäudes nötig.
- Die baulichen Maßnahmen zwischen verschiedenen Gebäuden sind geringer als bei einer Verkabelung.
- Höhere Mobilität, da theoretisch jeder Punkt eines Firmengeländes drahtlos erreichbar ist.

Nachteile

- Oft geringere Datenübertragungsraten als bei Kabeln, die abhängig von Hindernissen sind.
- Anfällig für Störeinflüsse und Abhören durch Unbefugte.
- Probleme mit Ausleuchtung und Reflexionen.
- Bei vielen gleichzeitigen Nutzern an einem WLAN-Zugang bricht die Übertragungsrate ein (Shared Media).

Sicherheit als kritischer Bereich

Gerade im Bereich Sicherheit gibt es bei WLANs einige Punkte zu beachten, die sich auch in einem etwas größeren Konfigurationsaufwand äußern. Die sogenannte **SSID (Service Set Identifier)** kann eine eindeutige Identifikation (Firmenname etc.) enthalten, damit bei Problemen eine Kontaktaufnahme mit dem Betreiber möglich ist. Ein Verbergen bringt nicht viel, da die SSID in jedem Paket mitgeschickt wird und es Programme gibt, die auch verborgene SSIDs auslesen können. Eine versehentliche Verbindung durch Unbefugte ist bei verschlüsselten Zugängen nicht zu erwarten. Es gibt auch Empfehlungen, als SSID eine zufällige Zeichenfolge einzugeben, damit die SSID keine Rückschlüsse auf den Betreiber zulässt.

Letztendlich sollte die Übertragung im WLAN nur **verschlüsselt** erfolgen, wobei die Verschlüsselung mit **WEP (Wired Equivalent Privacy)** unsicher ist. Besser ist der Einsatz von **WPA (Wi-Fi**

Protected Access) bzw. der Nachfolgetechnologie **WPA2**, da hier deutlich stärkere Verschlüsselungsmechanismen mit **AES (Advanced Encryption Standard)** verwendet werden. Zusammen mit **TKIP (Temporal Key Integrity Protocol)** sind allerdings nur max. 54 Mbit/s möglich! Höhere Raten erreicht z. B. WPA2 zusammen mit CCMP (Counter-Mode/CBC-Mac Protocol)

Grundlegende Beschreibung

Kommunikation über WLAN erfolgt entweder als Punkt-zu-Punkt- oder als Mehrpunkt-Kommunikation. Die erste Variante dient z. B. der Überwindung größerer Distanzen durch den Einsatz zweier Richtantennen.

Bei der Mehrpunkt-Kommunikation werden ein oder mehrere sogenannte Access Points eingesetzt, die im Prinzip jeweils wie Zentralen (Verteiler) fungieren und die Datenströme mehrerer Clients koordinieren.

Diese Access Points können bei größeren Installationen über Kabel und geeignete Managed Switches eine Verbindung zu einem sogenannten RADIUS (Remote Authentication Dial-In User Service)-Server erhalten, um zwischen berechtigten und nicht berechtigten Sendern zu unterscheiden (Authentifizierung).

Eine Unterscheidung über die MAC-Adresse sollte nur den zum Zugang berechtigten Geräten vorbehalten bleiben, die sich nicht per RADIUS authentifizieren können. Dabei wird in einer Zugangsliste (Access Control Table) vom Managed Switch die MAC-Adresse eingetragen. Auf einem Access-Point bringt dies keine höhere Sicherheit, da per Funk eine MAC-Adresse unverschlüsselt verschickt wird und somit gefälscht werden kann. Die kleinste Einheit ist eine sogenannte Funkzelle, womit der Bereich gemeint ist, der von einem Sender (=Access Point) abgedeckt werden kann. Er umfasst ca. 30m im Gebäude und bis zu 300m im Freien. Mit speziellen Antennen können auch mehrere Kilometer überbrückt werden.

ISM-Frequenzbänder

In den meisten Fällen wird von den Herstellern ein sogenanntes **ISM-Band (Industrial, Scientific and Medical)** verwendet. Manchmal finden Sie auch die Abkürzung ISMO, wobei der letzte Buchstabe für den Begriff „Office“ (Büro) steht. Der Einsatz dieser Frequenzbänder bietet zwei Vorteile. Sie sind

- **gebührenfrei**
- **genehmigungsfrei**

Hierin liegt aber auch gleichzeitig der Nachteil. Sie werden von sehr vielen Herstellern für die unterschiedlichsten Zwecke genutzt, wie z. B. drahtlose Lautsprecher, elektronische Türöffnung bei Autos oder Garagen, und so ist die Gefahr, dass sich Geräte gegenseitig stören, relativ hoch.

Die für WLAN wichtigsten ISM-Bänder sind

- das **2,4-GHz-Band** (2,3995 bis 2,4845 GHz) mit max. **13 überlappenden Kanälen** von 20 MHz Bandbreite; auch 40 MHz Bandbreite ist möglich, dann aber mit weit weniger nutzbaren Kanälen. Es sind Geschwindigkeiten bis zu 300MBit/s möglich.
- das **5-GHz-Band** (5,150 bis 5,350 GHz für Kanalnummer 36-64 und 5,470 bis 5,725 GHz für Kanalnummer 100-140) mit Kanälen von 20, 40, 80 oder 160 MHz Bandbreite, wobei **max. 19**

Kanäle bei 20 MHz Bandbreite nicht überlappend nutzbar sind. Beim Funken mit 40 MHz Bandbreite sind 2 dieser Kanäle gebündelt erforderlich, mit 80 MHz 4 Kanäle usw. Die **Reichweite ist geringer** als im 2,4 GHz-Band. Es sind Geschwindigkeiten bis zu 600MBit/s möglich.

- zukünftig das 60-GHz-Band (57 bis 66 GHz) mit vier 2000 MHz breiten Funkkanälen für kurze Distanzen

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4:4_02



Last update: **2020/01/19 15:48**

4.4) Ethernet

Erklärung

Ethernet ist eine Technik, die **Software (Protokolle usw.) und Hardware (Kabel, Verteiler, Netzwerkkarten usw.) für kabelgebundene Datennetze spezifiziert**, welche ursprünglich für lokale Datennetze (LANs) gedacht war und daher auch als LAN-Technik bezeichnet wird. Sie ermöglicht den Datenaustausch in Form von Datenframes zwischen den in einem lokalen Netz (LAN) angeschlossenen Geräten (Computer, Drucker und dergleichen). Derzeit sind Übertragungsraten von 1, 10, 100 Megabit/s (Fast Ethernet), 1000 Megabit/s (Gigabit-Ethernet), 2,5, 5, 10, 40, 50, 100, 200 und 400 Gigabit/s spezifiziert. In seiner ursprünglichen Form erstreckt sich das LAN dabei nur über ein Gebäude; Ethernet-Varianten über Glasfaser haben eine Reichweite von bis zu 70 km.

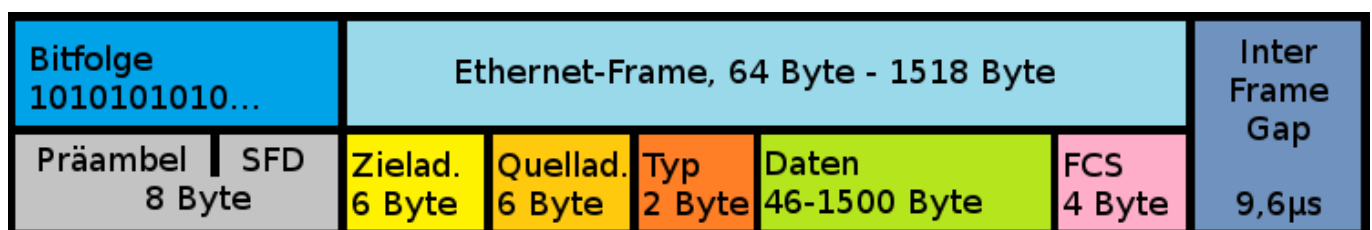
Die Ethernet-Protokolle umfassen Festlegungen für Kabeltypen und Stecker sowie für Übertragungsformen (Signale auf der Bitübertragungsschicht, Paketformate). Im OSI-Modell ist mit Ethernet sowohl die **physische Schicht (OSI Layer 1)** als auch die **Data-Link-Schicht (OSI Layer 2)** festgelegt.

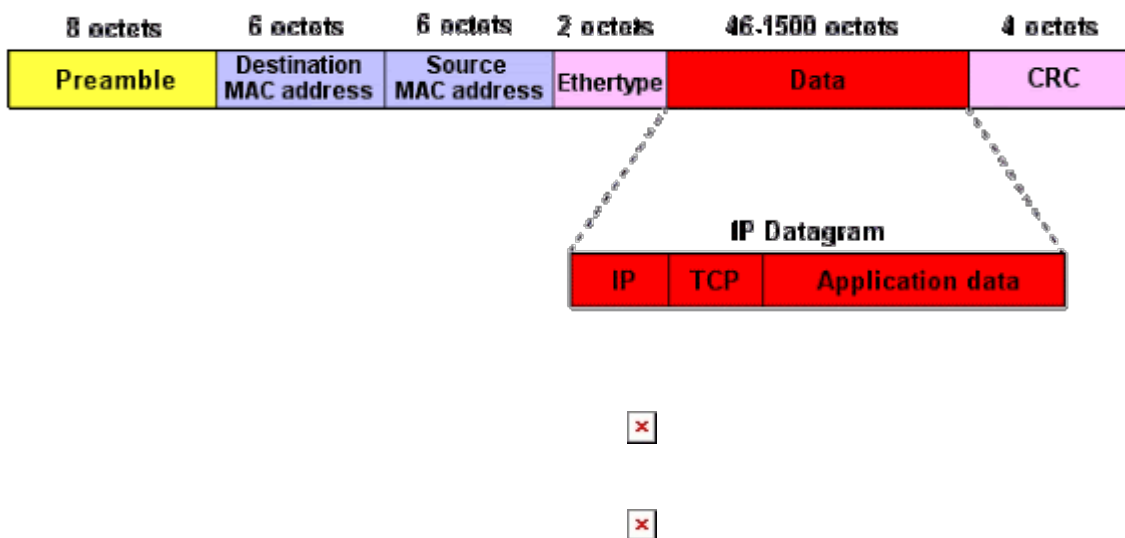
Ethernet basiert auf der Idee, dass die Teilnehmer eines LANs Nachrichten durch Hochfrequenz übertragen, allerdings **nur innerhalb eines gemeinsamen Leitungsnetzes**. Jede Netzwerkschnittstelle hat **einen global eindeutigen 48-Bit-Schlüssel**, der als **MAC-Adresse** (=Media-Access-Control-Adress) bezeichnet wird. Das stellt sicher, dass alle Systeme in einem Ethernet unterschiedliche Adressen haben.

Ethernet Frame

Bei der Übertragung von Daten über Ethernet ist das Ethernet-Frame hauptverantwortlich für die korrekte Regelsetzung und erfolgreiche Übermittlung von Datenpaketen. Versendete Daten über Ethernet werden vom Frame sozusagen getragen. Ein Ethernet-Frame ist zwischen 64 Byte und 1518 Byte groß, abhängig von der Größe der zu transportierenden Daten.

Im OSI-Modell befindet sich der Frame auf der Sicherungsschicht, die für die fehlerfreie Übertragung verantwortlich ist und trennt den Bitdatenstrom in Blöcke bzw. Frames auf.





Ein Ethernet-Frame muss standardmäßig mindestens 64 Byte groß sein, damit die Kollisionserkennung funktioniert, und kann maximal 1.518 Byte groß sein. Das Paket beginnt immer mit einer Präambel, die die Synchronisation zwischen Sender und Empfänger regelt und einem „Start Frame Delimiter“ (SFD), der das Frame definiert. Beide Informationen sind eine Bitfolge im Format 10101010... Im eigentlichen Frame finden sich Informationen zu Ziel- und Quelladressen (MAC-Format) und Steuerinformationen (im Fall von Ethernet II das Type-Field, später eine Längenangabe), dann folgt der zu übermittelnde Datensatz. Eine „Frame Check Sequence“ (FCS) schließt als Prüfsumme das gesamte Frame (ausgenommen Präambel und SFD). Das Paket wird von einem „Inter Frame Gap“ abgeschlossen, der eine 9,6 µs lange Sendepause festlegt.

Ethernet II benutzt die klassische Framestruktur, die das sogenannte Type-Field („Typ“) beinhaltet, womit verschiedene Protokolle der Vermittlungsschicht definiert werden. Im OSI-Modell ist die Vermittlungsschicht (auch „Network Layer“) wichtig für die Schaltung von Verbindungen und die Bereitstellung von Netzwerkadressen. Das Type-Field wurde in späteren Frame-Formaten durch eine Längenangabe ersetzt.

CSMA/CD

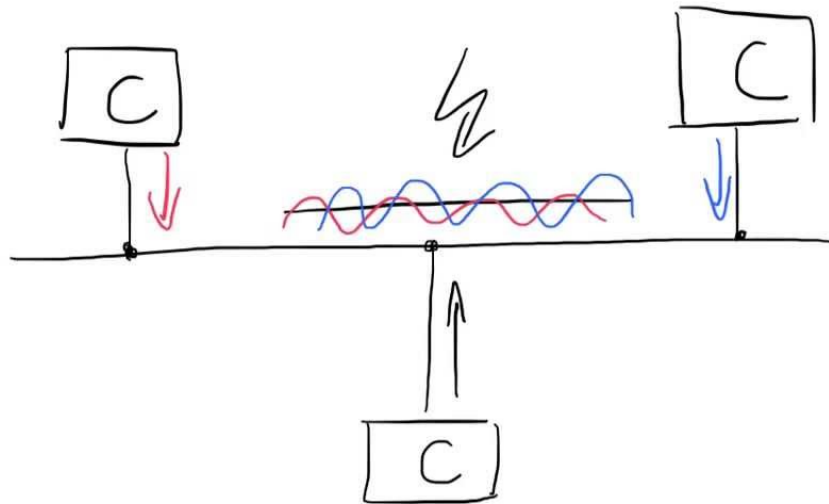
CSMA/CD (Carrier Sense Multiple Access with Collision Detection) ist das Zugriffsverfahren des Ethernets. Die Grundidee dabei ist, dass jede Station zu senden beginnen kann, wann sie will. Die einzelnen Stationen haben jederzeit und konkurrierend Zugang (Multiple Access) zum gemeinsamen Übertragungsmedium. Das grundlegende Motto könnte damit lauten:

Jeder darf, wann er will

Eingesetzt wird dieses Verfahren bei logischen Bus-Topologien. Dabei ist egal, ob physikalisch eine Bus- oder eine Stern-Topologie vorliegt.

Vorgehen

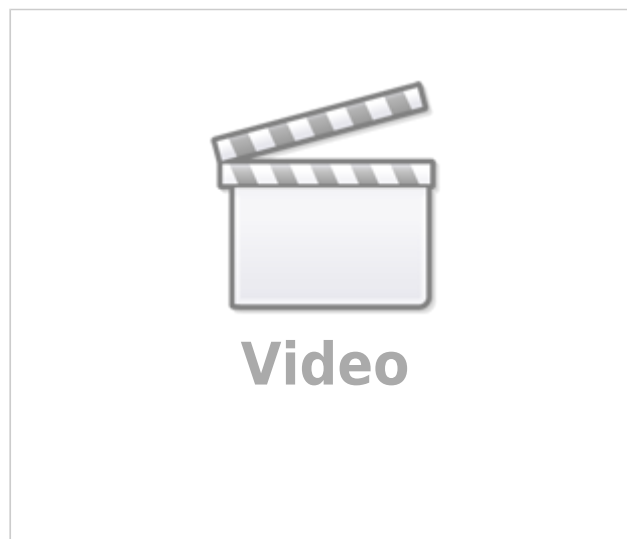
Durch Regelungen wird versucht, das Risiko zu minimieren, dass zwei Stationen ungünstigerweise gleichzeitig zu senden beginnen und somit die Signale auf dem Übertragungsweg zerstört/gestört werden (**=Kollision**).



1) Leitung prüfen (Carrier Sense)

Der erste Teil der Abkürzung steht für Kollisionsverhinderung. Dabei wird vor einer geplanten Sendung das Übertragungsmedium abgehört, ob dieses frei ist. Wenn das Medium frei ist, wird gesendet. -> **LISTEN BEFORE TALKING**

2) Erkennen von Kollisionen (Collision Detection) Kommt es trotzdem zu einer Kollision, weil zwei Stationen gleichzeitig zu senden beginnen (**Multiple Access**), dann muss diese Kollision erkannt (**Collision Detection**) und reagiert werden. Dabei müssen alle Stationen immer am Medium horchen, ob eine Kollision auftritt. Ist dies der Fall, so sendet die erste Station, die eine Kollision erkennt, ein sogenanntes JAM-Signal aus. Jede Station, die das JAM-Signal registriert, stoppt unmittelbar das Senden von Daten. Nach einer zufälligen Zeitspanne, wird das Medium wieder überprüft und anschließend wieder begonnen zu senden.



Frage: Wie kann bei einer physikalischen Stern-Topologie eine Kollision auftreten?

Vorteile

- Jeder kann zu jeder Zeit senden

Nachteile

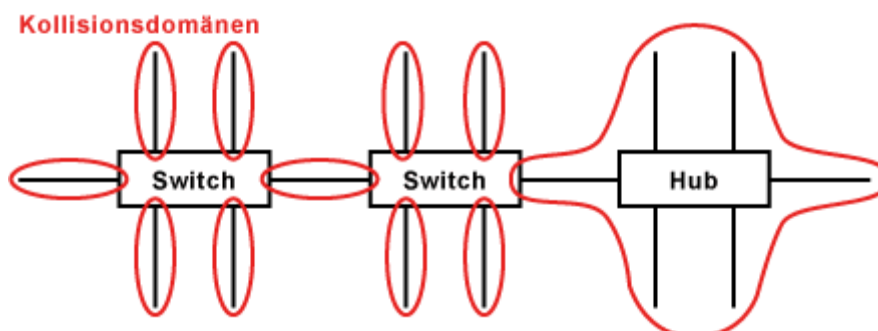
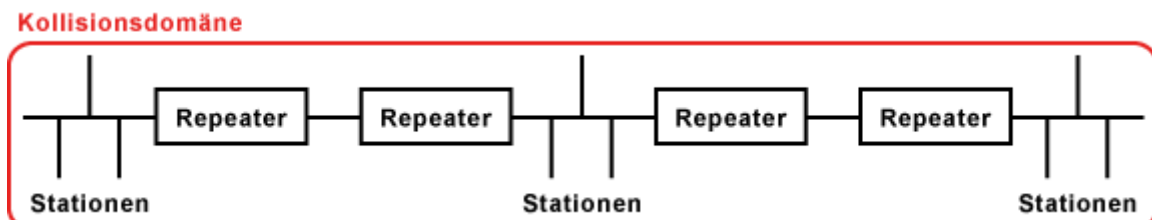
- Je mehr Stationen in einer Kollisionsdomäne, desto häufiger treten Kollisionen auf
- Der Zeitpunkt einer Sendung ist zufällig
- Das Verfahren ist ungeeignet für zeitkritischen Anwendungen

Kollisionsdomäne

Mit dem Begriff Kollisionsdomäne wird in einem Computernetz ein Teilbereich aus Teilnehmerstationen in derselben OSI-Modell-Schicht 1 bezeichnet. Eine Kollisionsdomäne umfasst alle Netzwerkgeräte, die um den Zugriff auf ein gemeinsames Übertragungsmedium konkurrieren. Das Übertragungsmedium ist daher eine zwischen allen Netzstationen geteilte Ressource. Grundlegende Vorstellung dabei ist, dass alle Netzwerkteilnehmer die Chance zur gleichberechtigten Nutzung des Netzwerkes besitzen.

Bei einem gemeinsamen Medium kann zu einer bestimmten Zeit nur jeweils eine Station Informationen übertragen, die an alle anderen Stationen übertragen bzw. von diesen empfangen wird. Fangen in einem derartigen gemeinsamen Schicht-1-Segment zwei Stationen gleichzeitig an zu senden, kommt es zu Kollisionen.

Beispiele für Kollisionsdomänen:



From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4:4_03



Last update: **2020/01/19 15:58**

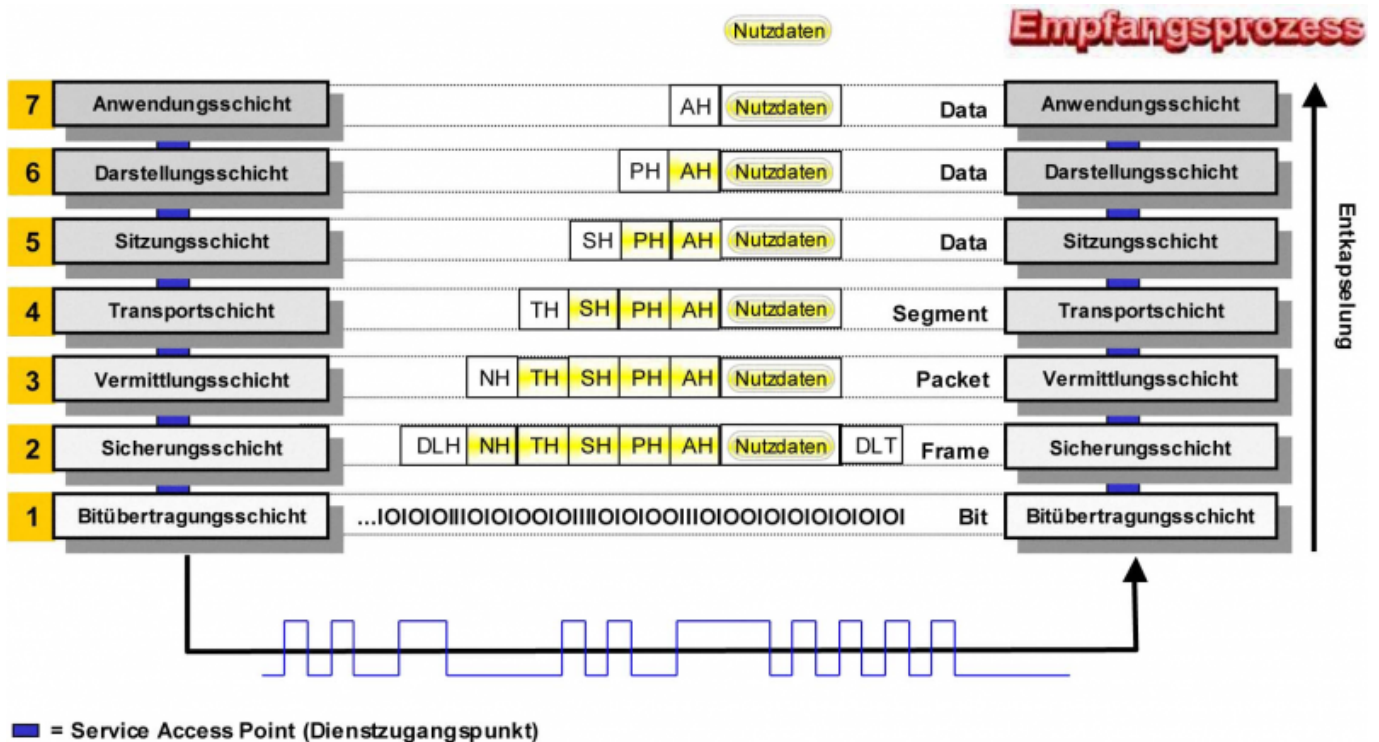
18.5) OSI - Schichtenmodell

Das OSI-7-Schichtenmodell ist ein Referenzmodell für herstellerunabhängige Kommunikationssysteme bzw. eine Design-Grundlage für Kommunikationsprotokolle und Computernetze. OSI steht für Open System Interconnection (Offenes System für Kommunikationsverbindungen) und wurde von der ISO (International Organization for Standardization), das ist die Internationale Organisation für Normung, als Grundlage für die Bildung von offenen Kommunikationsstandards entworfen. Bei allen ISO-Standards handelt es sich um Handlungsempfehlungen. Die Einhaltung einer ISO-Norm ist freiwillig. In der Regel wird die Einhaltung der ISO-Standards von verschiedenen Seiten, zum Beispiel Kooperationspartnern, Herstellern und Kunden, gefordert.



Das Modell

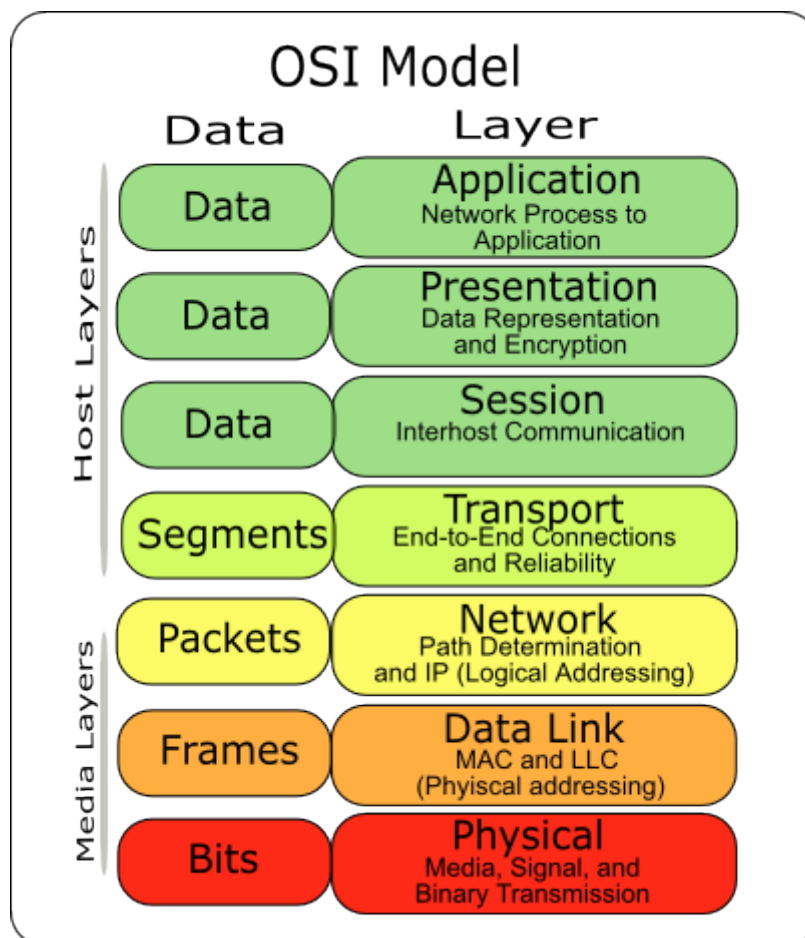
(Offenes System für Kommunikationsverbindungen) und wurde von der ISO (International Organization for Standardization), das ist die Internationale Organisation für Normung, als Grundlage für die Bildung von offenen Kommunikationsstandards entworfen. Bei allen ISO-Standards handelt es sich um Handlungsempfehlungen. Die Einhaltung einer ISO-Norm ist freiwillig. In der Regel wird die Einhaltung der ISO-Standards von verschiedenen Seiten, zum Beispiel Kooperationspartnern, Herstellern und Kunden, gefordert.



Die Schichten im Detail

Application Layer (Anwendungsschicht)	Benutzerschnittstelle, Dienste, Anwendungen und Netzmanagement
Schicht 7	Die Anwendungsschicht stellt Funktionen für die Anwendungen zur Verfügung. Diese Schicht stellt die Verbindung zu den unteren Schichten her. Auf dieser Ebene findet auch die Dateneingabe und -ausgabe statt.
Presentation Layer (Darstellungsschicht)	Übersetzung, Verschlüsselung , Kompression in Standardformate
Schicht 6	Die Darstellungsschicht setzt die Daten der Anwendungsebene in ein Zwischenformat um. Diese Schicht ist auch für Sicherheitsfragen zuständig. Durch sie werden Dienste zur Verschlüsselung von Daten bereitgestellt und gegebenenfalls Daten komprimiert.
Session Layer (Sitzungsschicht)	Erstellung einer Verbindung, Freigabe von Verbindungen, Dialogsteuerung
Schicht 5	Diese Schicht ermöglicht zwei Anwendungen auf verschiedenen Computern, eine gemeinsame Sitzung aufzubauen, damit zu arbeiten und sie zu beenden. Sie übernimmt ebenfalls die Dialogsteuerung zwischen den beiden Computern einer Sitzung und regelt, welcher der beiden wann und wie lange Daten überträgt.
Transport Layer (Transportschicht)	Logische Ende-zu-Ende-Verbindungen (Transportkontrolle, Paketbildung)
Schicht 4	Die Transportschicht stellt die zuverlässige Auslieferung der Nachrichten sicher und erkennt sowie behebt allfällige Fehler. Sie ordnet bei Bedarf auch die Nachrichten in Paketen neu, indem sie lange Nachrichten zur Datenübertragung in kleinere Pakete aufteilt. Am Ende des Weges stellt sie die kleinen Pakete wieder zur ursprünglichen Nachricht zusammen. Die empfangene Transportebene sendet auch eine Empfangs bestätigung.
Network Layer (Vermittlungsschicht)	Routing (Internet), Datenflusskontrolle, Adressierung

Application Layer (Anwendungsschicht)	Benutzerschnittstelle, Dienste, Anwendungen und Netzmanagement
Schicht 3	Die Vermittlungsschicht steuert die zeitliche und logische getrennte Kommunikation zwischen den Endgeräten, unabhängig vom Übertragungsmedium und der Topologie. Auf dieser Schicht erfolgt erstmals die logische Adressierung der Endgeräte. Die Adressierung ist eng mit dem Routing (Wegfindung vom Sender zum Empfänger) verbunden.
Data Link Layer (Sicherungsschicht)	Logische Verbindungen mit Datenpaketen und elementare Fehlererkennungsmechanismen
Schicht 2	Die Sicherungsschicht sorgt für eine zuverlässige und funktionierende Verbindung zwischen Endgerät und Übertragungsmedium. Zur Vermeidung von Übertragungsfehlern und Datenverlust enthält diese Schicht Funktionen zur Fehlererkennung, Fehlerbehebung und Datenflusskontrolle. Auf dieser Schicht findet auch die physikalische Adressierung von Datenpaketen statt.
Physical Layer (Physikalische Schicht)	Maßnahmen und Verfahren zur Übertragung von Bitfolgen
Schicht 1	Die Bitübertragungsschicht definiert die elektrische, mechanische und funktionale Schnittstelle zum Übertragungsmedium. Die Protokolle dieser Schicht unterscheiden sich nur nach dem eingesetzten Übertragungsmedium und -verfahren. Das Übertragungsmedium ist jedoch kein Bestandteil der Schicht 1.



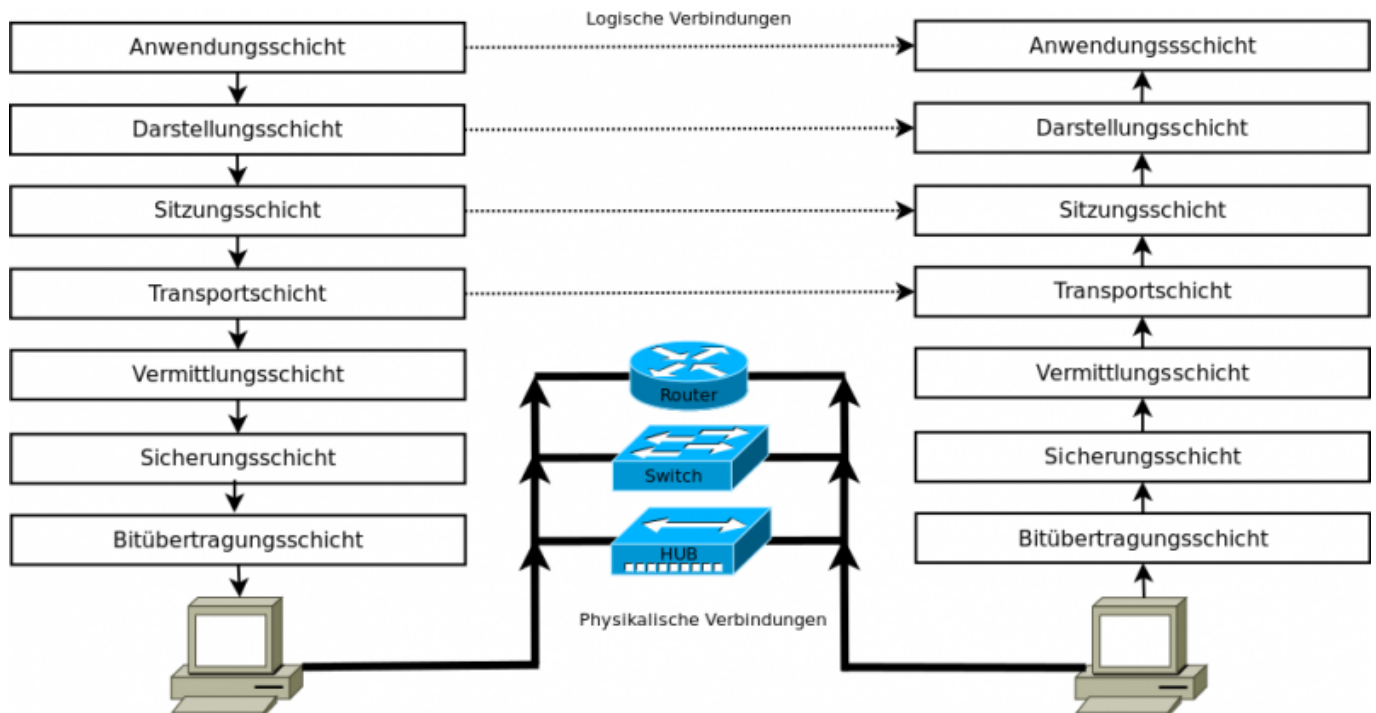
Protokolle

Protokolle sind eine **Sammlung von Regeln zur Kommunikation** auf einer bestimmten Schicht des OSI-Schichtenmodells. Die Protokolle einer Schicht sind zu den Protokollen der über- und untergeordneten Schichten weitestgehend transparent, so dass die Verhaltensweise eines Protokolls sich wie bei einer direkten Kommunikation mit dem Gegenstück auf der Gegenseite darstellt.

Die **Übergänge zwischen den Schichten sind Schnittstellen**, die von den Protokollen verstanden werden müssen. Weil manche Protokolle für ganz bestimmte Anwendungen entwickelt wurden, kommt es auch vor, dass sich **Protokolle über mehrere Schichten** erstrecken und mehrere Aufgaben abdecken. Dabei kommt es vor, dass in manchen Verbindungen einzelne Aufgaben in mehreren Schichten und somit mehrfach ausgeführt werden.

	OSI-Schichtenmodell	TCP/IP-Stack Protokolle	Einheiten	Netzwerkopplung	
Upper Layers Anwendungsorientiert	7 Application-Layer Anwendungs-Schicht	Anwendungs-Stack Dienste(Protokolle (Auswahl)): HTTP (Hypertext Transfer Protocol) / HTTPS FTP (File Transfer Protocol) SMTP (Simple Mail Transfer Protocol) POP (Post Office Protocol) DNS (Domain Name System) NFS (Network File System) SMB (Server Message Block) XMPP (Extensible Messaging and Presence Pr.) LDAP (Lightweight Directory Access Protocol)	Daten	Gateway, Content-Switch, Proxy	End-to-End / Multihop
	6 Presentation-Layer Darstellungs-Schicht				
	5 Session-Layer Sitzungs-Schicht				
Transport Service Transportorientiert	4 Transport-Layer Transport-Schicht	Transport-Stack TCP, SCTP (verbindungsorientiert) UDP (verbindungslos) TLS (oberhalb TCP)	TCP: Segmente UDP: Datagramme		Point-to-Point
	3 Network-Layer Vermittlungs-Schicht	Internet-Stack IP, IPsec, ICMP (verbindungslos)	Pakete max. 64 kByte	Router, Layer3-Switch	
	2 Data-Link-Layer Sicherungs-Schicht	Netzzugang-Stack Ethernet, TokenRing, FDDI, MAC, ARCnet	Rahmen, Frame max. 1518 Byte oder 1522 Byte mit VLAN-Tag 9000 Byte Jumboframe	Bridge, Switch WirelessAccessPoint	
	1 Physical-Layer Bitübertragungs-Schicht		Bit's Symbole Pakete	Netzkabel, Hub, Repeater	
		Siehe auch MTU (MaximalTransmissionUnit)		ARP	

Anmerkung zur Skizze: End-zu-End-Verbindungen finden laut Definition erst ab Schicht 4 statt.



Das wichtigste Protokoll im Netzwerkverkehr, ist das **TCP & IP Protokoll** (Transmission Control Protocol & Internet Protocol), welche sich heute als Standard durchgesetzt haben.

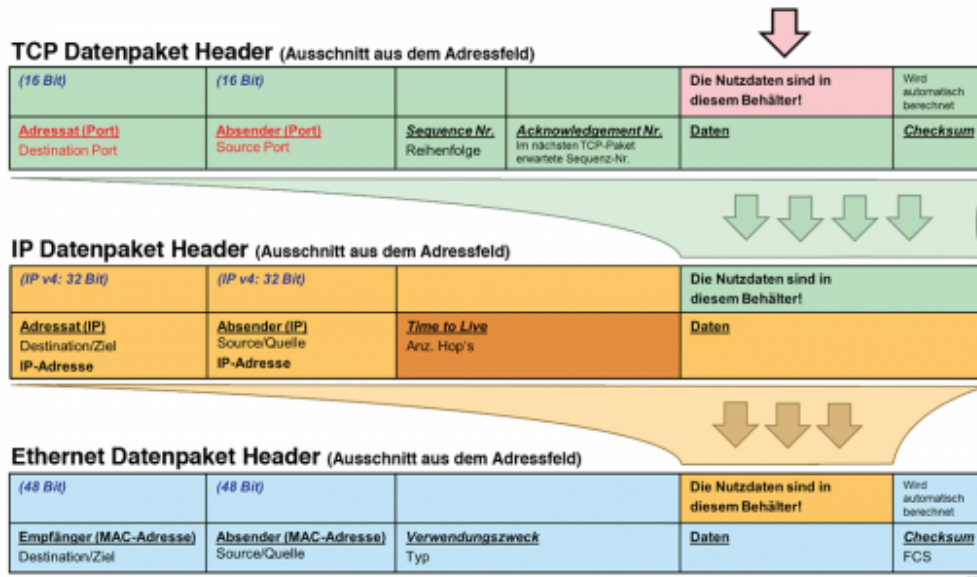
Neben TCP und IP gibt es natürlich noch viele weitere Protokolle, wie in der obigen Abbildung zu sehen ist.

Die folgende Tabelle listet einige dieser Protokolle auf und ordnet sie ins obige Modell ein:

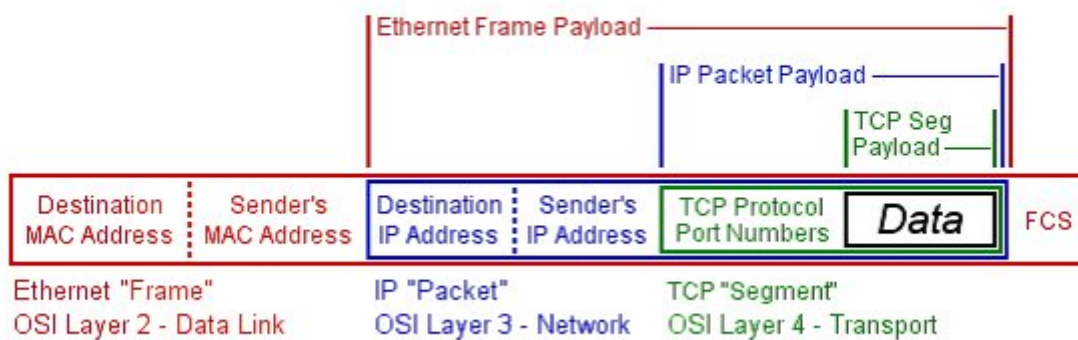
Protokoll	Schicht	Name	Beschreibung
FTP	5	File Transfer Protocol	Datenaustausch zwischen Rechnern
Telnet	5	Telecommunication Network Protocol	Terminalemulation zur Host-Kommunikation
SMTP	5	Simple Mail Transfer Protocol	Versenden von E-Mails
HTTP	5	Hypertext Transfer Protocol	Übertragen von HTML-Seiten
POP	5	Post Office Protocol	Abrufen von E-Mails
TCP	4	Transmission Control Protocol	Aufbau logischer Verbindungen zwischen Applikationen
UDP	4	User Datagram Protocol	Verbindungsloses Übertragungsprotokoll. Es ist nicht so gesichert wie TCP dafür aber schneller
IP	3	Internet Protocol	Verbindungsloses Protokoll zur Paketlenkung und Paketvermittlung über IP-Adressen
IPSec	3	IP Secure	Erweitert das reguläre IP-Protokoll um ein Bündel von Sicherheitsmechanismen
ARP	3	Address Resolution Protocol	Dien dazu logische IP-Adressen physikalischen MAC-Adressen zuzuordnen

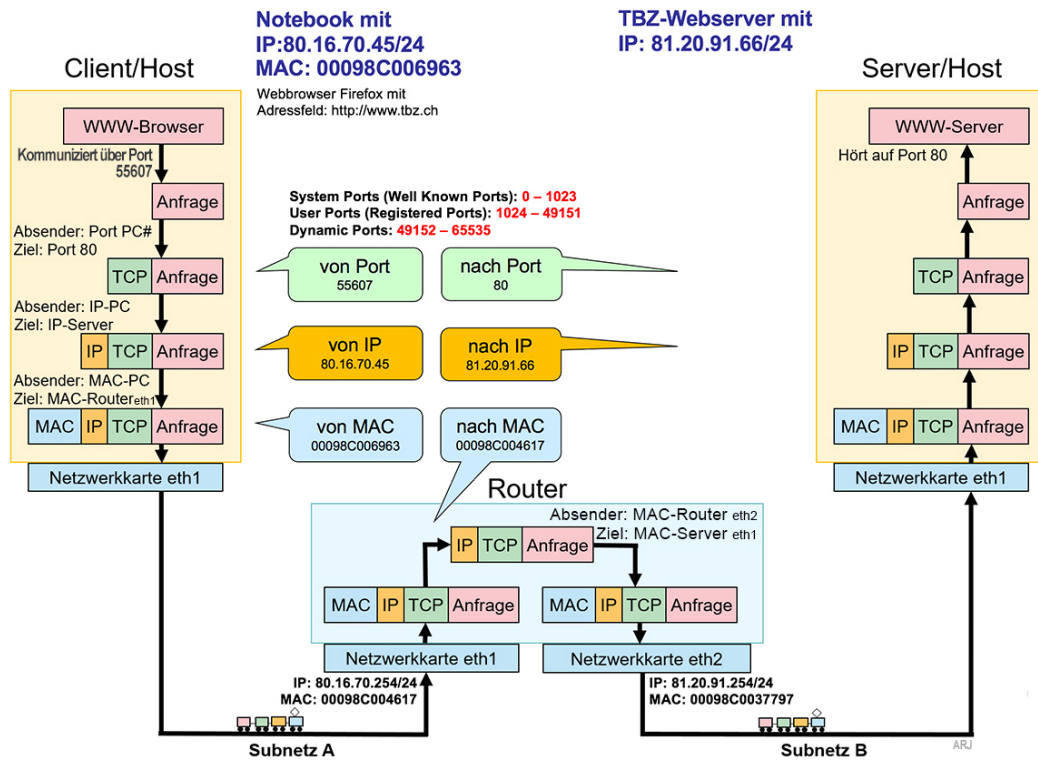
Datenkapselung

Unter der Datenkapselung versteht man den Prozess im OSI-Modell und im TCP/IP-Referenzmodell, der die zu versendenden Daten im Header (und ggf. Trailer) der jeweiligen Schichten ergänzt. Im OSI-Modell betrifft dies die Datagramme der Schichten 2 bis 4, die gekapselt (verpackt) werden.



Encapsulation Payloads





OSI-Schichtenmodell

TCP/IP-Referenzmodell

https://edu.juergarnold.ch/fach_it/netzwerktheorie/article1.html?q=osi#nw_3_19

From:
<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4:4_04

Last update: **2020/01/19 16:11**



Netzwerkkomponenten

In der Netzwerktechnik unterscheidet man zwischen aktiven und passiven Netzwerk-Komponenten. Während **aktive Netzwerk-Komponenten eine eigene Logik haben**, zählen die passiven Netzwerk-Komponenten zur fest installierten Netzwerk-Infrastruktur. In der Regel dienen Netzwerk-Komponenten zur Kopplung der Netzwerk-Stationen. Man spricht deshalb auch von Kopplungselementen.

Passive Netzwerk-Komponenten

- Patchkabel und Installationskabel
- Anschlussdose
- Steckverbinder
- Patchfeld / Patchpanel
- Netzwerk-Schrank / Patch-Schrank

Hinweis: Zu den passiven Netzwerk-Komponenten zählen die Bestandteile der Verkabelung. Diese ist im OSI-Schichtenmodell nicht definiert.

Aktive Netzwerk-Komponenten

In kleinen privaten Netzwerken, haben Netzwerk-Komponenten noch klare Bezeichnung, wie Switch oder Router. In großen Unternehmensnetzwerken ist die Benennung der Kopplungselemente nicht immer eindeutig.

- Netzwerkkarte
- Repeater
- Hub
- Bridge
- Switch
- Router
- Gateway
- Server

Netzwerkkarte

Eine Netzwerkkarte wird auch als Netzwerkadapter bezeichnet. Die englische Bezeichnung ist Network Interface Card (NIC). Eine Netzwerkkarte ermöglicht es, auf ein Netzwerk zuzugreifen und arbeitet auf der Bitübertragungsschicht (Schicht 1) und der Datumsicherungsschicht (Schicht 2) des OSI-Schichtenmodells. Jede Netzwerkkarte hat eine Hardware-Adresse (Format: XX-XX-XX-XX-XX-XX), die es auf der Welt nur einmal gibt. Anhand dieser Adresse lässt sich eine Station auf der Bitübertragungsschicht adressieren.

Im Falle von Ethernet-Netzen besteht die **MAC-Adresse aus 48 Bit (sechs Bytes)**. Die Adressen werden in der Regel **hexadezimal** geschrieben. Üblich ist dabei eine **byteweise Schreibweise**,

wobei die einzelnen Bytes durch Bindestriche oder Doppelpunkte voneinander getrennt werden, z. B. 00-80-41-ae-fd-7e oder 00:80:41:ae:fd:7e. Seltener zu finden sind Angaben wie 008041aefd7e oder 0080.41ae.f7

In den **ersten 24 Bits** (Bit 3 bis 24) wird eine von der IEEE vergebene **Herstellerkennung** (auch OUI – Organizationally Unique Identifier genannt) beschrieben, die weitgehend in einer Datenbank einsehbar sind[6]. Die **verbleibenden 24 Bits** (Bit 25 bis 48) werden **vom jeweiligen Hersteller** für jede Schnittstelle individuell **festgelegt**.

Repeater

Ein Repeater ist ein Kopplungselement, um die Übertragungsstrecke innerhalb von Netzwerken, zum Beispiel Ethernet, zu verlängern. Ein Repeater empfängt ein Signal und bereitet es neu auf. Danach sendet er es weiter. Auf diese Weise verlängert der Repeater die Übertragungsstrecke und räumliche Ausdehnung des Netzwerks. Im einfachsten Fall hat ein Repeater zwei Ports, die wechselweise als Ein- und Ausgang funktionieren (bidirektional).

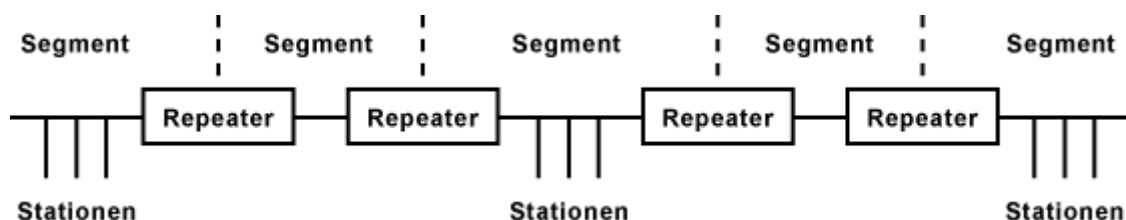
Repeater versteht man in der Regel als Verstärker von Übertragungsstrecken. Die weitere Beschreibung bezieht sich auf Repeater in kabelgebundenen Netzwerken, speziell in Ethernet-Netzwerken.

Ein Repeater arbeitet auf der Schicht 1, der Bitübertragungsschicht des OSI-Schichtenmodells. Der Repeater übernimmt keinerlei regulierende Funktion in einem Netzwerk. Er kann nur Signale empfangen und weiterleiten. Für angeschlossene Geräte ist nicht erkennbar, ob sie an einem Repeater angeschlossen sind. Er verhält sich völlig transparent.

Ein Repeater erweitert somit eine Kollisionsdomäne!!

Ein Repeater mit mehreren Ports wird auch als Hub (Multiport-Repeater) bezeichnet. Er kann mehrere Netzwerk-Segmente miteinander verbinden.

Die Repeater-Regel (5-4-3)



Um ein großes Netzwerk mit einer möglichst großen Reichweite aufzubauen, können mehrere Repeater hintereinandergeschaltet werden. Allerdings, nicht in beliebiger Anzahl. Der Grund liegt im Laufzeitverhalten und der Phasenverschiebung zwischen den Signalen an den Enden des Netzwerks. Deshalb gilt folgende Repeater-Regel:

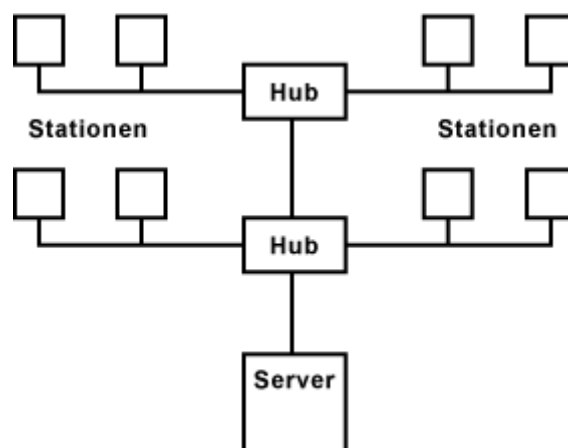
Es dürfen nicht mehr als fünf (5) Kabelsegmente verbunden werden. Dafür werden vier (4) Repeater eingesetzt. An nur drei (3) Segmenten dürfen Endstationen angeschlossen werden.

Diese **Repeater-Regel** hat nur in den **Ethernet-Netzwerken 10Base2 und 10BASE5** eine

Bedeutung. In Netzwerken, die mit Switches und Router aufgebaut sind, hat diese Repeater-Regel keine Bedeutung. Um die Nachteile von Repeatern in Ethernet-Netzwerken zu umgehen, werden generell Switches zur Kopplung der Hosts eingesetzt. In großen Netzwerken, insbesondere über unterschiedliche Übertragungssysteme hinweg, werden zusätzlich Router eingesetzt.

Hub

Ein Hub ist ein Kopplungselement, das mehrere Hosts in einem Netzwerk miteinander verbindet. In einem Ethernet-Netzwerk, das auf der Stern-Topologie basiert, dient ein Hub als Verteiler für die Datenpakete. Hubs arbeiten auf der Bitübertragungsschicht (Schicht 1) des OSI-Schichtenmodells und sind damit auf die **reine Verteilfunktion** beschränkt. **Hubs erweitern somit die Kollisionsdomäne.** Ein Hub nimmt **ein Datenpaket entgegen und sendet es an alle anderen Ports** weiter. Das bedeutet, er **broadcastet**. Dadurch sind nicht nur **alle Ports**, sondern **auch alle Hosts belegt**. Sie bekommen alle Datenpakete zugeschickt, auch wenn sie nicht die Empfänger sind. Für die Hosts bedeutet das auch, dass sie nur dann senden können, wenn der Hub gerade keine Datenpakete sendet. Sonst kommt es zu Kollisionen.



Wenn die Anzahl der Anschlüsse an einem Hub für die Anzahl der Hosts nicht ausreicht, dann benötigt man noch einen zweiten Hub. Zwei Hubs werden über einen Uplink-Port eines der beiden Hubs oder mit einem Crossover-Kabel (Sende- und Empfangsleitungen sind gekreuzt) verbunden. Es gibt auch spezielle „stackable“ Hubs, die sich herstellerspezifisch mit Buskabeln kaskadieren lassen. Durch die Verbindung mehrerer Hubs lässt sich die Anzahl der möglichen Hosts im Netzwerk erhöhen. Allerdings ist die Anzahl der anschließbaren Hosts begrenzt. Hier gilt die Repeater-Regel.

Nachteile

- ineffizient
- unsicher (Jeder bekommt jede Nachricht)

Vorteile

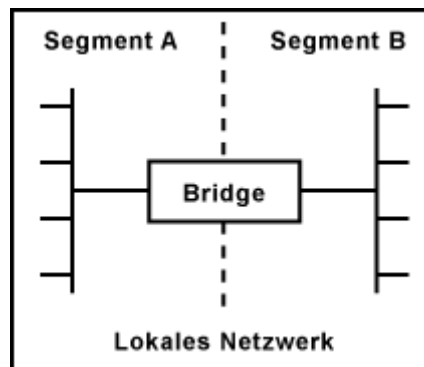
- zentraler Verteiler

Bridge

Eine Bridge ist ein Kopplungselement, das ein lokales Netzwerk in zwei Segmente aufteilt. Dabei

werden die Nachteile von Ethernet, die besonders bei großen Netzwerken auftreten ausgeglichen. Als Kopplungselement ist die Bridge eher untypisch. Man vermeidet die Einschränkungen durch Ethernet heute eher durch Switches.

Eine Bridge teilt eine Kollisionsdomäne!!

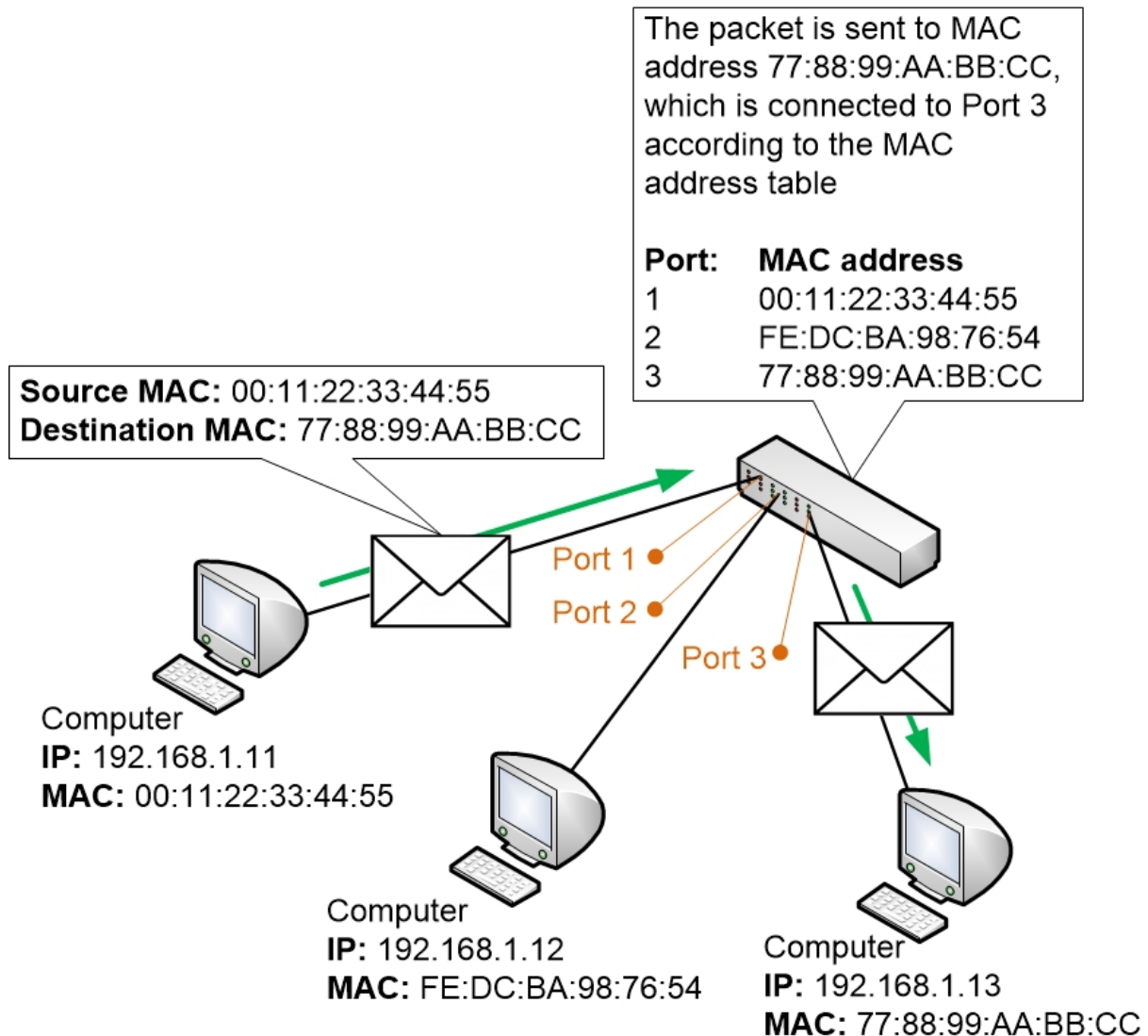


Switch

Ein Switch ist ein Kopplungselement, das mehrere Stationen in einem Netzwerk miteinander verbindet. In einem Ethernet-Netzwerk, das auf der Stern-Topologie basiert, dient ein Switch als Verteiler für die Datenübertragung.

Die Funktion ist ähnlich einem Hub, mit dem Unterschied, dass ein Switch direkte Verbindungen zwischen den angeschlossenen Geräten schalten kann, sofern ihm die Ports der Datenpaket-Empfänger bekannt sind. Somit kommunizieren wirklich nur jene miteinander, die auch miteinander reduzieren wollen. Wenn nicht, dann broadcastet der Switch die Datenpakete an alle Ports. Wenn die Antwortpakete von den Empfängern zurück kommen, dann merkt sich der Switch die MAC-Adressen der Datenpakete und den dazugehörigen Port und sendet die Datenpakete dann nur noch dorthin. Er baut also eine sogenannte MAC-Adressen-Tabelle auf:

Beispiel:



Während ein Hub die Bandbreite des Netzwerks limitiert, steht der Verbindung zwischen zwei Hosts, die volle Bandbreite der Ende-zu-Ende-Netzwerk-Verbindung zur Verfügung.

Ein Switch arbeitet auf der Sicherungsschicht (Schicht 2) des OSI-Modells und arbeitet ähnlich wie eine Bridge. Daher haben sich bei den Herstellern auch solche Begriffe durchgesetzt, wie z. B. Bridging Switch oder Switching Bridge. Die verwendet man heute allerdings nicht mehr.

Switches unterscheidet man hinsichtlich ihrer Leistungsfähigkeit mit folgenden Eigenschaften:

- Anzahl der speicherbaren MAC-Adressen für die Quell- und Zielports
- Verfahren, wann ein empfangenes Datenpaket weitervermittelt wird (Switching-Verfahren)
- Latenz (Verzögerungszeit) der vermittelten Datenpakete

Ein Switch ist im Prinzip nichts anderes als ein intelligenter Hub, der sich merkt, über welchen Port welcher Host erreichbar ist. Teure Switches können zusätzlich auf der Schicht 3, der Vermittlungsschicht, des OSI-Schichtenmodells arbeiten (Layer-3-Switch oder Schicht-3-Switch). Sie sind in der Lage, die Datenpakete anhand der IP-Adresse an die Ziel-Ports weiterzuleiten. Im Gegensatz zu normalen Switches lassen sich auch ohne Router logische Abgrenzungen erreichen.

Switching-Verfahren

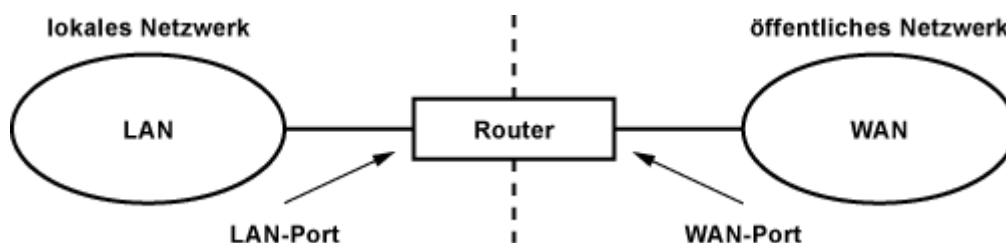
Switching-Verfahren	Beschreibung	Vorteile	Nachteile
Cut-Through	Der Switch leitet das Datenpaket sofort weiter, wenn er die Adresse des Ziels erhalten hat.	Die Latenz, die Verzögerungszeit, zwischen Empfangen und Weiterleiten ist äußerst gering.	Fehlerhafte Datenpakete werden nicht erkannt und trotzdem an den Empfänger weitergeleitet.
Store-and-Forward	Der Switch nimmt das gesamte Datenpaket in Empfang und speichert es in einem Puffer. Dort wird dann das Paket mit verschiedenen Filtern geprüft und bearbeitet. Erst danach wird das Paket an den Ziel-Port weitergeleitet.	Fehlerhafte Datenpakete können so im voraus aussortiert werden.	Die Speicherung und Prüfung der Datenpakete verursacht eine Verzögerung, abhängig von der Größe des Datenpaketes.
Fragment-Free	Der Switch empfängt die ersten 64 Byte des Daten-Paketes. Ist dieser Teil fehlerlos werden die Daten weitergeleitet. Die meisten Fehler und Kollisionen treten während den ersten 64 Byte auf. Dieses Verfahren wird trotz seiner effektiven Arbeitsweise selten genutzt.	sehr effizient	selten genutzt

Router

Ein Router verbindet mehrere Netzwerke mit unterschiedlichen Protokollen und Architekturen. Ein Router befindet sich häufig an den Außengrenzen eines Netzwerks, um es mit dem Internet oder einem anderen, größeren Netzwerk zu verbinden. Über die Routing-Tabelle entscheidet ein Router, welchen Weg ein Datenpaket nimmt. Es handelt sich dabei um ein dynamisches Verfahren, das Ausfälle und Engpässe ohne den Eingriff eines Administrators berücksichtigen kann. Ein Router hat mindestens zwei Netzwerkanschlüssen. Er arbeitet auf der Vermittlungsschicht (Schicht 3) des OSI-Schichtenmodells.

Die Aufgabe eines Routers ist ein komplexer Vorgang, der sich in 4 Schritte einteilen lässt:

- Ermittlung der verfügbaren Routen
- Auswahl der geeignetsten Route unter Berücksichtigung verschiedener Kriterien
- Herstellen einer physikalischen Verbindung zu anderen Netzwerken
- Anpassen der Datenpakete an die Übertragungstechnik (Fragmentierung)



Ein Router hat in der Regel zwei Anschlüsse. Einen für die LAN-Seite und einen für die WAN-Seite.

Häufig sind die Ports mit der Bezeichnung LAN und WAN gekennzeichnet. Manchmal gibt es Port-Beschriftungen, bei denen nicht immer eindeutig ist, um was es sich handelt. Mit LAN ist immer das lokale Netzwerk mit privaten IP-Adressen gemeint, während die WAN-Seite das öffentliche Netzwerk kennzeichnet.



Zusammenfassung



Hub, Switch & Router

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4:4_05



Last update: **2020/02/11 09:21**

IP-Adressen

Damit die Wegwahl in Netzwerken und im Internet zur Übermittlung von Datenpaketen vom Senden zum Empfänger funktioniert, wird jedem Rechner im **Internet eine weltweit eindeutige (und einmalige) Adresse (IP-Adresse)** zugeordnet.

Für Rechner die ohne Router direkt mit dem Internet verbunden sind, heißt das, dass deren IP-Adressen weltweit eindeutig sind.

Standardmäßig haben diese IP-Adressen eine Länge von **32 Bit (4 x 8Bit)**.

Diese werden aus Gründen der **Übersichtlichkeit in 4 Zahlen zu je 1 Byte** aufgeteilt. Meist werden die einzelnen Bytes durch einen Punkt getrennt und dezimal dargestellt.

Öffentliche IP-Adressen

Wenn man von öffentlichen Adressen spricht, meint man die IP-Adressen, die im Internet erreichbar sind. Die Zuweisung einer öffentlichen IP-Adresse erfolgt in der Regel durch einen Provider (z.B. A1, Telekom, ...) welche diese wiederum in Europa durch die Organisation [RIPE-NCC](#) bekommt.



Private IP-Adressen

Private IP-Adressen (abgekürzt Private IP) sind IP-Adressen, die von der **IANA** nicht im Internet vergeben sind. Sie wurden für die **private Nutzung aus dem öffentlichen Adressraum ausgespart**, damit sie ohne administrativen Mehraufwand (Registrierung der IP-Adressen) in lokalen Netzwerken genutzt werden können. Als die **IP-Adressen des Internet Protokolls v4 knapp** wurden und dadurch eine **bewusste Einsparung öffentlicher IP-Adressen** notwendig wurde, war es umso wichtiger, private IP-Adressen in lokalen Netzwerken zur Verfügung zu haben, die **beliebig oft bzw. in beliebigen Netzwerken genutzt** werden können.

Netzklasse: Anzahl Netze (ohne Subnetting)	Netzadressbereich	Subnetzmaske	CIDR-Notation	Anzahl Adressen
Klasse A: 1 privates Netz mit 16.777.216 Adressen	10.0.0.0 bis 10.255.255.255	255.0.0.0	10.0.0.0/8	$2^{24} = 16.777.216$
Klasse B: 16 private Netze mit jeweils 65.536 Adressen	172.16.0.0 bis 172.31.255.255	255.240.0.0	172.16.0.0/12	$2^{20} = 1.048.576$
Klasse C: 256 private Netze mit jeweils 256 Adressen	192.168.0.0 bis 192.168.255.255	255.255.0.0	192.168.0.0/16	$2^{16} = 65.536$

Loopback Adresse

Die Class-A-Netzwerkadresse 127 ist weltweit reserviert für das sogenannte local loopback; sie dient zu Testzwecken der Netzwerkschnittstelle des eigenen Rechners. Die IP-Adresse **127.0.0.1** ist standardmäßig dem Loopback-Interface jedes Rechners zugeordnet.

Alle an diese Adresse geschickten Datenpakete werden nicht nach außen ins Netzwerk gesendet, sondern an der Netzwerkschnittstelle reflektiert. Die Datenpakete erscheinen, als kämen sie aus einem angeschlossenen Netzwerk.

Vergleich IP-Adresse vs. Telefonnummer

Ähnlich wie bei einer Telefonnummer setzt sich eine IP-Adresse aus mehreren Segmenten zusammen. Eine Telefonnummer besteht aus einer Vorwahl und einer Teilnehmernummer. Führen Sie ein Ortsgespräch, so muss die Vorwahl nicht angegeben werden. Ähnlich ist es bei IP-Adressen.

Diese bestehen auch aus zwei Teilen, wobei die ID für Identifikation steht:

- **Netzwerk-ID** im vorderen linken Teil entspricht der Vorwahl und gibt das entsprechende IP-Subnetz an.
- **Host-ID** im hinteren rechten Teil kennzeichnet eine einzelne Netzwerkkarte und entspricht dem Teilnehmernummer im Ortsnetz.

Entsprechend können Rechner im selben Subnetz direkt miteinander kommunizieren. Dagegen erfordert Kommunikation zwischen Subnetzen eine Vermittlungsstelle, einen Router (Standardgateway), wo alle nicht im selben Netz adressierten Pakete hingeschickt werden.

Um zu erkennen, wo die Netzwerk-ID endet und die Host-ID beginnt, muss zusätzlich zur IP-Adresse zwingend eine sogenannte Subnetzmaske mit angegeben werden.

Subnetzmaske

Eine Subnetzmaske ist ein Bitmuster, das (von links nach rechts) Teile der IP-Adresse „maskiert“, um den Übergang zwischen Netz-ID und Host-ID zu kennzeichnen. Binär betrachtet besteht eine Subnetzmaske aus einer Folge von Einsen, die ab einer bestimmten Stelle umschlägt in eine Folge von Nullen. Dieser Umschlagpunkt gibt an, wie viele Bits zur Netzwerk-ID (Einsen) und zur Host-ID (Nullen) gehören.

Ein Auszug aus den möglichen Subnetzmasken für ein Klasse C-Netz

Maske	Maske (kurze Schreibweise)	Anzahl Hosts pro Netz	Netze	Beispiel Klasse C-Netz (192.168.1.0)
255.255.255.0	/24	256	1	192.168.1.0 - 192.168.1.255
255.255.255.128	/25	128	2	192.168.1.0 - 192.168.1.127 192.168.1.128 - 192.168.1.255

Maske	Maske (kurze Schreibweise)	Anzahl Hosts pro Netz	Netze	Beispiel Klasse C-Netz (192.168.1.0)
255.255.255.192	/26	64	4	192.168.1.0 - 192.168.1.63 192.168.1.64 - 192.168.1.127 192.168.1.128 - 192.168.1.191 192.168.1.192 - 192.168.1.255
255.255.255.224	/27	32	8	192.168.1.0 - 192.168.1.31 192.168.1.32 - 192.168.1.63 ... 192.168.1.192 - 192.168.1.223 192.168.1.224 - 192.168.1.255
255.255.255.240	/28	16	16	192.168.1.0 - 192.168.1.15 192.168.1.16 - 192.168.1.31 ... 192.168.1.224 - 192.168.1.239 192.168.1.240 - 192.168.1.255
255.255.255.248	/29	8	32	192.168.1.0 - 192.168.1.7 192.168.1.8 - 192.168.1.15 ... 192.168.1.240 - 192.168.1.247 192.168.1.248 - 192.168.1.255
255.255.255.252	/30	4	64	192.168.1.0 - 192.168.1.3 192.168.1.4 - 192.168.1.7 ... 192.168.1.248 - 192.168.1.251 192.168.1.252 - 192.168.1.255
255.255.255.254	/31	2	128	192.168.1.0 - 192.168.1.1 192.168.1.2 - 192.168.1.3 ... 192.168.1.252 - 192.168.1.253 192.168.1.254 - 192.168.1.255
255.255.255.255	/32	1	256	192.168.1.0 192.168.1.1 ... 192.168.1.254 192.168.1.255



Erklärung

Subnetting

Die Aufteilung eines zusammenhängenden Adressraums von IP-Adressen in mehrere kleinere Adressräume nennt man Subnetting. Ein Subnet, Subnetz bzw. Teilnetz ist ein physikalisches Segment eines Netzwerks, in dem IP-Adressen mit der gleichen Netzwerkadresse benutzt werden. Diese Teilnetze können über Routern miteinander verbunden werden und bilden dann ein großes zusammenhängendes Netzwerk.

Beispiel 1:

IP-Adresse: 192.168.128.17

Dezimale Darstellung:	192.	168.	128.	17
Bit-Darstellung:	1100 0000	1010 1000	1000 0000	0001 0001
Hex-Darstellung:	C0	A8	80	11

Subnetzmaske: 255.255.255.0

Dezimale Darstellung:	255.	255.	255.	0
Bit-Darstellung:	1111 1111	1111 1111	1111 1111	0000 0000
Hex-Darstellung:	FF	FF	FF	00

Verknüpft man nun die binäre IP-Adresse mit der Subnetzmaske mit einem Logischen AND, so bekommt man die Netz-ID (=Netzadresse).

	<----- NETZ-ID ----->				<- HOST-ID ->		
	1100 0000	1010 1000	1000 0000	0001 0001			(IP-Adresse)
AND	1111 1111	1111 1111	1111 1111	0000 0000			(Subnetzmaske)

	1100 0000	1010 1000	1000 0000	0000 0000			(Netz-ID)
Netz-ID in Dezimaldarstellung:							
192.168.128							

Verknüpft man nun die binäre IP-Adresse mit der negierten Subnetzmaske mit einem Logischen AND, so bekommt man die Host-ID.

	<----- NETZ-ID ----->				<- HOST-ID ->		
	1100 0000	1010 1000	1000 0000	0001 0001			(IP-Adresse)
AND	0000 0000	0000 0000	0000 0000	1111 1111			(negierte Subnetzmaske)

-							
	0000 0000	0000 0000	0000 0000	0001 0001			(Host-ID)
Host-ID in Dezimaldarstellung:							
17							

Das heißt im Netzwerk 192.168.128.0 stehen theoretisch 256 (0-255) Adressen zur Adressierung von Netzwerkgeräten zur Verfügung. Praktisch sind es aber nur 254 Adressen, da zwei Adressen

- die **Netzadresse** (= 1. Adresse im Netz -> 192.168.128.0)
- die **Broadcastadresse** (= letzte Adresse im Netz -> 192.168.128.255)

reserviert sind.

Beispiel 2:

IP-Adresse 130.94.122.195/27

	Dezimal	Binär				
Berechnung						
IP Adresse	130.094.122.195	10000010	01011110	01111010	11000011	
ip-adresse						
Netzmaske	255.255.255.224	11111111	11111111	11111111	11100000	AND
netzmaske						

Netzwerkteil	130.094.122.192	10000010	01011110	01111010	11000000	=
netzwerkanteil						
IP Adresse	130.094.122.195	10000010	01011110	01111010	11000011	
ip-adresse						
Netzmaske	255.255.255.224	00000000	00000000	00000000	00011111	AND(NOT
netzmaske)						

Geräteteil		3	00000000	00000000	00000000	=
geräteteil						

Bei einer Netzmaske mit 27 gesetzten Bits ergibt sich ein Netzwerkteil von 130.94.122.192. Es verbleiben 5 Bits und damit $2^5=32$ Adressen für den Geräteteil. Hiervon werden noch je 1 Adresse für das Netzwerk selbst und für den Broadcast benötigt, sodass 30 Adressen für Geräte zur Verfügung stehen.

Beispiel 3:

IP-Adresse 130.94.122.117/28

	Dezimal	Binär				
Berechnung						
IP Adresse	130.094.122.117	10000010	01011110	01111010	01110101	
ip-adresse						
Netzmaske	255.255.255.240	11111111	11111111	11111111	11110000	AND
netzmaske						

Netzwerkteil	130.094.122.112	10000010	01011110	01111010	01110000	=
netzwerkanteil						

IP Adresse	130.094.122.117	10000010	01011110	01111010	01110101	
ip-adresse						
Netzmaske	255.255.255.240	00000000	00000000	00000000	00001111	AND(NOT
netzmaske)						

Geräteteil	0. 0. 0. 5	00000000	00000000	00000000	00000101	=
geräteteil						

Bei einer Netzmaske mit 28 gesetzten Bits ergibt sich ein Netzwerkteil von 130.94.122.112. Es verbleiben 4 Bits und damit $2^4=16$ Adressen für den Geräteteil. Hiervon werden noch je 1 Adresse für das Netzwerk selbst und für den Broadcast benötigt, sodass 14 Adressen für Geräte zur Verfügung stehen.

- [Zusammenfassung zu Subnetzmasken](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4:4_06



Last update: **2020/02/15 20:53**

Übungen IP-Adressierung

Bsp 1

Befinden sich 192.168.0.93/27 und 192.168.0.97/27 im gleichen Netzwerk-Segment?

[Lösung](#)

```
1) NW-ID: 192.168.0.64  
2) NW-ID: 192.168.0.96  
=> Nein
```

Bsp 2

Wie lauten die Netzwerkadressen des ersten/letzten Hosts des Netzwerkes 192.168.0.96/27 und die Broadcastadresse?

[Lösung](#)

```
1. Host: 192.168.0.97  
letzter Host: 192.168.0.126  
BC: 192.168.0.127
```

Bsp 3

Das Netz 195.1.31.0 /24 soll in weitere 30 Subnetze aufgeteilt werden. Bestimme die Subnetzmaske, und die Anzahl Adressen pro Subnet!

[Lösung](#)

```
Subnetzmaske: 255.255.255.248  
Adressen: 6
```

Bsp 4

Das Netz 10.0.0.0 /16 soll in mindestens 200 Subnetze weiter unterteilt werden. Bestimme die Subnetzmaske, und die Anzahl Adressen pro Subnet!

[Lösung](#)

```
Subnetzmaske: 255.255.255.0
```

Adressen: 256

Bsp 5

Das Netz 192.168.1.0 /24 soll in Subnetze mit je 18 Host-Adressen aufgeteilt werden. Bestimme die Subnetzmaske, und die Anzahl Subnetze!

Lösung

Subnetzmaske: 255.255.255.224
Subnetze: 8

Bsp 6

Das Netz 192.168.100.0 /24 soll in Subnetze mit je 5 Host-Adressen aufgeteilt werden. Bestimmen Sie die Subnetzmaske, und die Anzahl Subnetze!

Lösung

Subnetzmaske: 255.255.255.248
Subnetze: 32

Bsp 7

Bestimme Netz- und Broadcastadresse des Subnets, in dem die Adresse 195.1.31.135 mit Netzmaske 255.255.255.128 liegt!

Lösung

NW-ID: 195.1.31.128
BC: 195.1.31.255

Bsp 8

Bestimme Netz- und Broadcastadresse des Subnets in dem die Adresse 195.1.31.135 mit Netzmaske 255.255.255.192 liegt!

Lösung

NW-ID: 195.1.31.128/26
BC: 195.1.31.191

Bsp 9

Bestimme Netz- und Broadcastadresse des Subnets in dem die Adresse 15.3.128.222 mit Netzmaske 255.255.255.240 liegt!

Lösung

NW-ID: 15.3.128.208
Broadcast: 15.3.128.223

Bsp 10

Gib den Bereich der Rechneradressen an, den das Teilnetz Nummer drei (132.45.96.0/19) hat.

Lösung

132.45.96.1
bis
132.45.127.254

Bsp 11

Berechne Netz-, Broadcast und Hostadressen des 0., 1., 15. und 31 Subnetz des Netzes 192.168.1.0/29

Lösung

0. Netz:
* NW-ID: 192.168.1.0
* BC: 192.168.1.7
* Hosts: 192.168.1.1-6

1. Netz:
* NW-ID: 192.168.1.8
* BC: 192.168.1.15
* Hosts: 192.168.1.9-14

15. Netz:
* NW-ID: 192.168.1.120
* BC: 192.168.1.127
* Hosts: 192.168.1.121-126

31. Netz:
* NW-ID: 192.168.1.248
* BC: 192.168.1.255
* Hosts: 192.168.1.249-254

Bsp 12

Vervollständige folgende Tabelle:

IP-Adresse	10.1.1.1	Netzwerkadresse des Subnet	...
Netmask	255.255.0.0	Broadcastadresse des Subnet	...
Anzahl Host pro Subnet		Anzahl Subnets in diesem Netz	

Lösung

IP-Adresse	10.1.1.1	Netzwerkadresse des Subnet	10.1.0.0
Netmask	255.255.0.0	Broadcastadresse des Subnet	10.1.255.255
Anzahl Host pro Subnet	65536	Anzahl Subnets in diesem Netz	256

Bsp 13

Vervollständige folgende Tabelle:

IP-Adresse	10.1.1.1/24	Netzwerkadresse des Subnet	...
Netmask	255.255.255.0	Broadcastadresse des Subnet	...
Anzahl Host pro Subnet		Anzahl Subnets in diesem Netz	

Lösung

IP-Adresse	10.1.1.1/24	Netzwerkadresse des Subnet	10.1.1.0
Netmask	255.255.255.0	Broadcastadresse des Subnet	10.1.1.255
Anzahl Host pro Subnet	254	Anzahl Subnets in diesem Netz	65536

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4:4_06:4_06_1



Last update: **2020/02/18 09:50**

IP-Routing

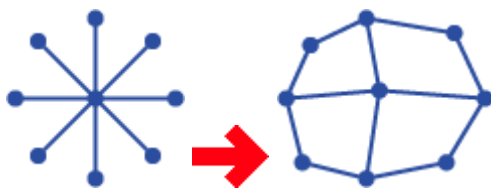
Das **Internet Protocol (IP)** ist ein **routingfähiges Protokoll** und sorgt dafür, dass **Datenpakete über Netzgrenzen** hinweg einen Weg zu anderen Hosts finden. Es kann die Daten über jede Art von physikalischer Verbindung oder Übertragungssystem vermitteln. Der hohen Flexibilität steht ein hohes Maß an Komplexität bei der Wegfindung vom Sender zum Empfänger gegenüber. Der **Vorgang der Wegfindung wird Routing** genannt

Wozu Routing?

Das grundlegende Verbindungselement in einem Ethernet-Netzwerk ist der Hub oder Switch. Daran sind alle Netzwerk-Teilnehmer angeschlossen. Wenn ein Host Datenpakete verschickt, dann werden die Pakete im Hub an alle Stationen verschickt und von diesen angenommen. Jedoch verarbeitet nur der adressierte Host die Pakete weiter. Das bedeutet aber auch, dass sich alle Hosts die Gesamtbandbreite dieses Hubs teilen müssen.

Um die Nachteile von Ethernet in Verbindung mit CSMA/CD auszuschließen, wählt man als Kopplungselement einen Switch und nutzt Fast Ethernet (kein CSMA/CD mehr). Der Switch merkt sich die Hardware-Adressen (MAC-Adressen) der Stationen und leitet die Ethernet-Pakete nur an den Port, hinter dem sich die Station befindet. Ist einem Switch die Hardware-Adresse nicht bekannt, leitet er das Datenpaket an alle seine Ports weiter (Broadcast) und funktioniert in diesem Augenblick wie ein Hub. Neben der begrenzten Speichergröße des Switches machen sich viele unbekannte Hardware-Adressen negativ auf die Performance eines Netzwerks bemerkbar.

Daher eignet sich zum **Verbinden großer Netzwerke** weder ein Hub noch ein Switch. Aus diesem Grund wird ein Netzwerk **durch Router** und IP-Adressen in **logische Segmente bzw. Subnetze** unterteilt. Die Adressierung durch das Internet Protocol ist so konzipiert, dass der Netzwerkverkehr innerhalb der Subnetze bleibt und erst dann das Netzwerk verlässt, wenn das Ziel in einem anderen Netzwerk liegt.



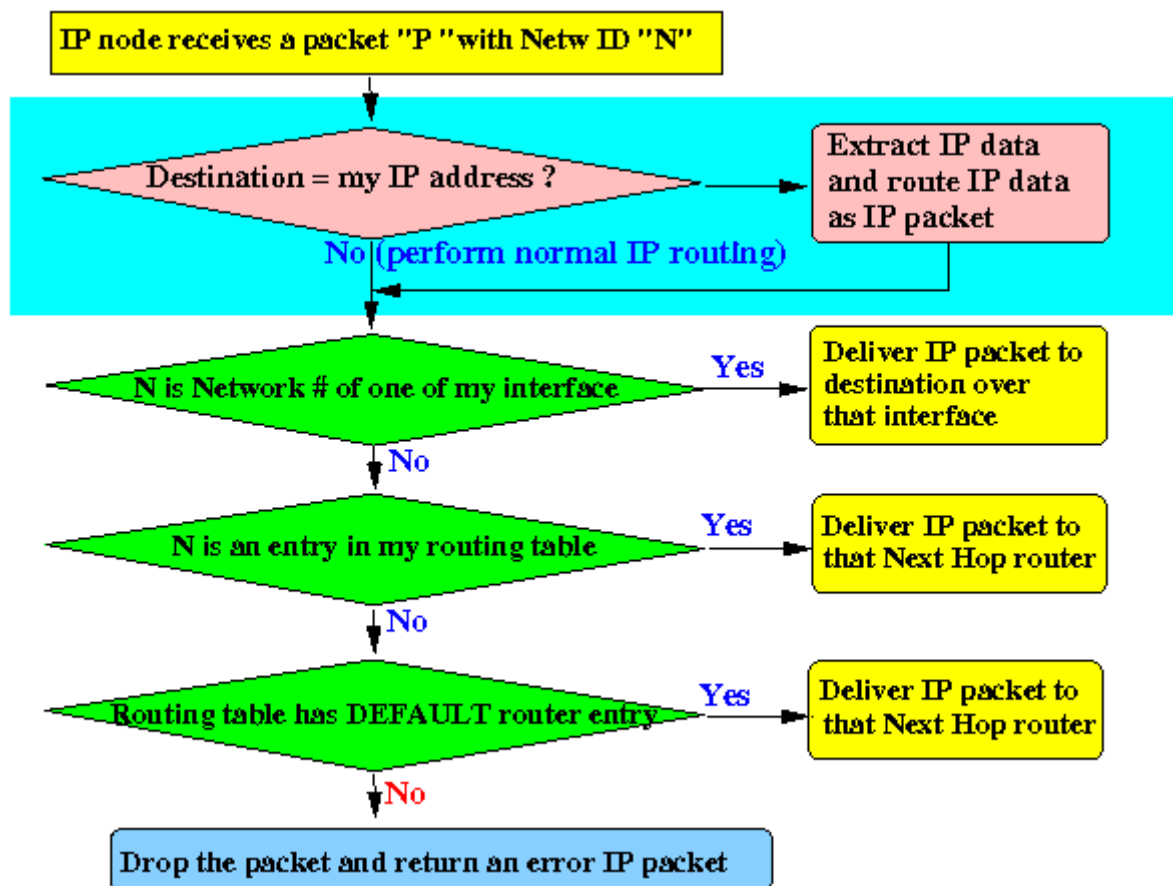
Insbesondere folgende **Probleme** in einem Ethernet-Netzwerk machen **IP-Routing notwendig**:

- **Vermeidung von Kollisionen und Broadcasts durch Begrenzung der Kollisions- und Broadcastdomäne**
- **Routing über unterschiedliche Netzarchitekturen und Übertragungssysteme**
- **Paket-Filter und Firewall**
- **Routing über Backup-Verbindungen bei Netzausfall**

IP-Routing-Algorithmus

Der IP-Routing-Algorithmus gilt nicht nur für IP-Router, sondern für alle Host, die IP-Datenpakete

empfangen können. Die empfangenen Datenpakete durchlaufen diesen Algorithmus bis das Datenpaket zugeordnet oder weitergeleitet werden kann.



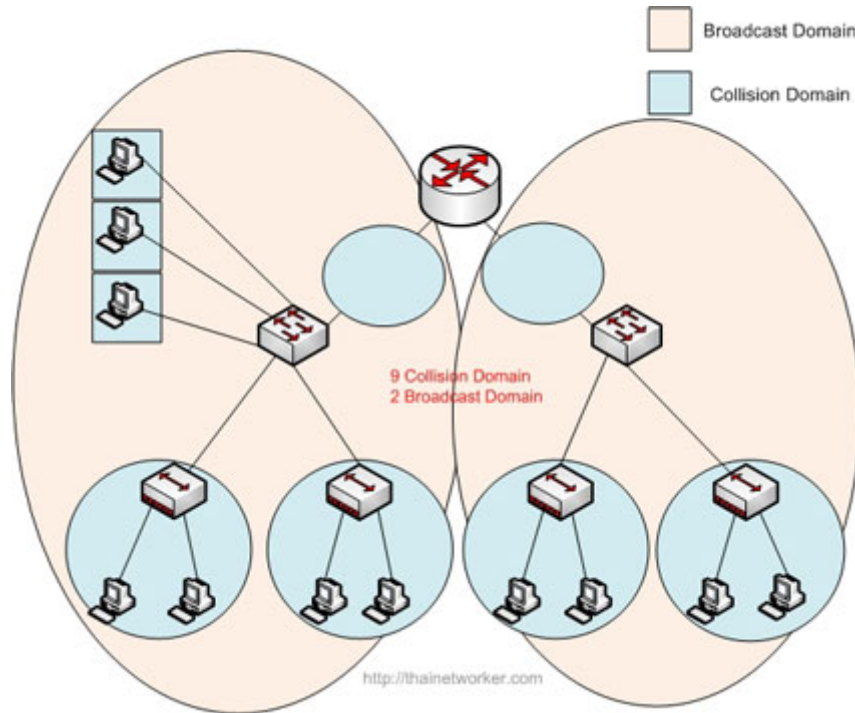
- Prüfe, ob das Datenpaket mir gehört?
 - Wenn ja, dann hat das Datenpaket sein Ziel erreicht und kann verarbeitet werden.
 - Wenn nein, prüfe ob das Datenpaket in mein Subnetz gehört?
 - Wenn ja, schick es in das eigene Subnetz weiter!
 - Wenn nein, prüfe ob die Route zum Empfänger bekannt ist?
 - Wenn ja, schicke es über die bekannte Route zum nächsten Router!
 - Wenn nein, prüfe ob es ein Standard-Gateway gibt?
 - Wenn ja, schick das Paket zum Standard-Gateway!
 - Wenn nein, schreibe eine Fehlermeldung und verwirf das Datenpaket!

Broadcastdomäne

Eine Broadcast-Domäne ist ein logischer Verbund von Netzwerkgeräten in einem lokalen Netzwerk, der sich dadurch auszeichnet, dass ein **Broadcast alle Domänenteilnehmer erreicht**.

Ein lokales Netzwerk auf der 2. Schicht des OSI-Modells (Sicherungsschicht) besteht durch seine Hubs, Switches und/oder Bridges aus einer Broadcast-Domäne. Erst durch die Unterteilung in VLANs oder durch den Einsatz von Routern, die auf Schicht 3 arbeiten, wird die Broadcast-Domäne aufgeteilt.

Eine Broadcast-Domäne besteht aus einer oder mehreren Kollisionsdomänen.



From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4:4_07



Last update: **2020/02/16 20:53**

Netzwerkbefehle

Windows-Kommandos

Wichtige Netzwerkbefehle unter Windows XP

- ipconfig
 - /all ... detaillierte Informationen über Netzwerk-Konfiguration
 - /renew ... erneuert IP-Adressen
 - /release ... gibt IP-Adressen frei
- ping (sendet Datenpakete zu Rechner)
 - ping IP-Adresse
 - ping hostname
 - -n Anzahl ... Anzahl der Pakete
- tracert (Route zu Rechner)
- nslookup
 - nslookup (DNS-Abfragen)
 - nslookup IP-Adresse
 - nslookup Domain-Name

Linux-Kommandos

- ifconfig

Zeigt Informationen zu Netzwerk-Interfaces

- ping

```
ping -c 4 www.example.com
```

Pingt 4 mal www.example.com

- tracepath

```
tracepath www.example.com
```

Zeigt Hops zum Host an (benötigt eventuell SU-Rechte)

```
sudo tracepath www.example.com
```

- nslookup

```
nslookup www.example.com
```

Überprüft DNS-Eintrag

weitere hilfreiche Befehle:

- nmap (scannt („mappt“) das Netzwerk, führt Portscans aus und findet die Software eines fremden PCs heraus)
- mtr (kombiniert tracer (ohne su-Rechte) und ping, anschauliche Darstellung)

Fragen - Aufgaben - Arbeitsaufträge

Arbeitsaufträge:

1. Gib deiner/m Nachbarn/in die IP-Adresse deines Computers und notiere dir die IP-Adresse seines/ihrer Computers.
2. Versuche deinen Nachbarcomputer anzupingen. Wie lange brauchte das Datenpaket zu diesem Rechner?
3. Gib deine IP-Adresse frei und versuche den Nachbarn anzupingen bzw. dich anpingen zu lassen. Erneure anschließend deine IP-Adresse und überprüfe, wie sie nun lautet.
4. Versuche deinen Rechner (mit deiner IP-Adresse und der Loopback-Adresse) anzupingen.
5. Finde heraus, welche IP-Adresse der Domain-Name mail.bgamstetten.ac.at hat.
6. Finde einen Rechner im Internet, bis zu welchem ein Datenpaket sehr lange dauert und schicke zu diesem Rechner 20 Datenpakete hintereinander.
7. Finde heraus, welcher Domain-Name zu folgender IP-Adresse gehört 131.130.250.250
8. Lasse dir die Route zu www.yahoo.com anzeigen. Wie viele Rechner liegen zwischen dir und dem Webserver von www.yahoo.com ?
9. Versuche mit dem Internetexplorer die Website von www.google.at aufzurufen, indem du die IP-Adresse von Google in der Browserzeile eingibst.
10. Suche im Internet die Homepage von der australischen Regierung. Welche IP-Adresse hat der Rechner, auf dem die Homepage liegt? Wie viele Rechner liegen zwischen dir und diesem Rechner?
11. Finde einen Rechner im Internet, bis zu dem möglichst viele Hops dazwischen sind (Wer findet die meisten?).

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4:4_08

Last update: **2020/02/16 20:53**



Protokolle

Protokolle sind eine **Sammlung von Regeln zur Kommunikation** auf einer bestimmten Schicht des OSI-Schichtenmodells. Die Protokolle einer Schicht sind zu den Protokollen der über- und untergeordneten Schichten weitestgehend transparent, so dass die Verhaltensweise eines Protokolls sich wie bei einer direkten Kommunikation mit dem Gegenstück auf der Gegenseite darstellt.

Die **Übergänge zwischen den Schichten sind Schnittstellen**, die von den Protokollen verstanden werden müssen. Weil manche Protokolle für ganz bestimmte Anwendungen entwickelt wurden, kommt es auch vor, dass sich **Protokolle über mehrere Schichten** erstrecken und mehrere Aufgaben abdecken. Dabei kommt es vor, dass in manchen Verbindungen einzelne Aufgaben in mehreren Schichten und somit mehrfach ausgeführt werden.

Protokoll-Stack

Da sich ein einzelnes Protokoll immer nur um eine Teilaufgabe im Rahmen der Kommunikation kümmert, werden mehrere Protokolle zu Protokollsammlungen oder Protokollfamilien, den sogenannten Protokoll-Stacks, zusammengefasst. Die wichtigsten Einzel-Protokolle werden dann oft stellvertretend als Bezeichnung des gesamten Protokoll-Stapels genutzt.

Kommunikation zwischen Netzwerkkomponenten funktioniert nur dann, wenn sie denselben Protokoll-Stack benutzen oder wenn Geräte eingesetzt werden, die zwischen verschiedenen Stacks vermitteln können.

Portnummern (ab Layer 5)

Um die einzelnen Dienste (Protokolle), die bei einem Rechner über dieselbe IP-Adresse ausgeführt werden, voneinander zu differenzieren, wurden Portnummern eingeführt, um bei einer Anfrage deutlich zu machen, welcher Dienst gemeint ist.

Diese Portnummern, die im TCP- oder UDP Header angegeben werden, sind weltweit eindeutig festgelegt und können z.B. auf der Homepage der [IANA](http://iana.org) eingesehen werden.

Allgemein lässt sich also sagen, dass die **IP-Adresse** den Rechner, **und** Die **Port-Nummer** den Dienst auf dem jeweiligen Rechner angibt. Diese beiden Informationen zusammen werden als **Socket** bezeichnet.

Wichtige Protokolle

OSI-LAYER	PROTOKOLL (Port)
5-7	DHCP (67+68/UDP)
	DNS (53/UDP)
	HTTP (80/TCP)
	HTTPS (443/TCP)
	FTP (20+21/TCP)
	SSH (22/TCP)
	SMTP (465 und 587 oder 25/TCP)
	POP3 (995 oder 110/TCP)
	IMAP (993 oder 143/TCP)
4	TCP
	UDP
3	IPv4
	IPv6
	NAT
	ICMP
2	Ethernet 802.3 , Wireless 802.x, MAC, ISDN,...
1	

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4:4_09



Last update: **2020/03/17 07:41**

Ethernet

Erklärung

Ethernet ist eine Technik, die **Software (Protokolle usw.) und Hardware (Kabel, Verteiler, Netzwerkkarten usw.) für kabelgebundene Datennetze spezifiziert**, welche ursprünglich für lokale Datennetze (LANs) gedacht war und daher auch als LAN-Technik bezeichnet wird. Sie ermöglicht den Datenaustausch in Form von Datenframes zwischen den in einem lokalen Netz (LAN) angeschlossenen Geräten (Computer, Drucker und dergleichen). Derzeit sind Übertragungsraten von 1, 10, 100 Megabit/s (Fast Ethernet), 1000 Megabit/s (Gigabit-Ethernet), 2,5, 5, 10, 40, 50, 100, 200 und 400 Gigabit/s spezifiziert. In seiner ursprünglichen Form erstreckt sich das LAN dabei nur über ein Gebäude; Ethernet-Varianten über Glasfaser haben eine Reichweite von bis zu 70 km.

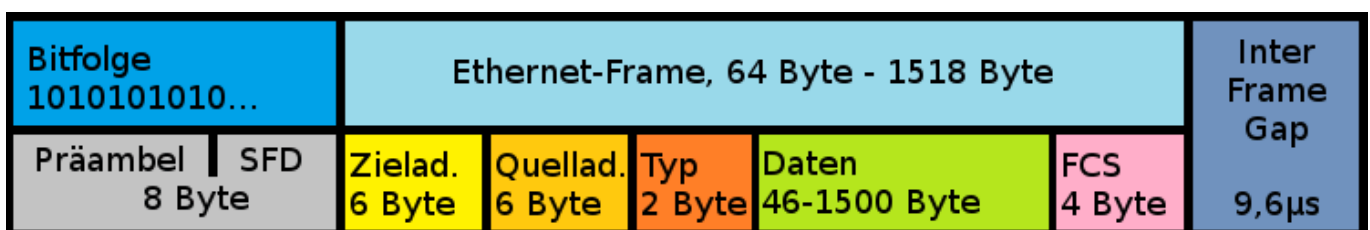
Die Ethernet-Protokolle umfassen Festlegungen für Kabeltypen und Stecker sowie für Übertragungsformen (Signale auf der Bitübertragungsschicht, Paketformate). Im OSI-Modell ist mit Ethernet sowohl die **physische Schicht (OSI Layer 1)** als auch die **Data-Link-Schicht (OSI Layer 2)** festgelegt.

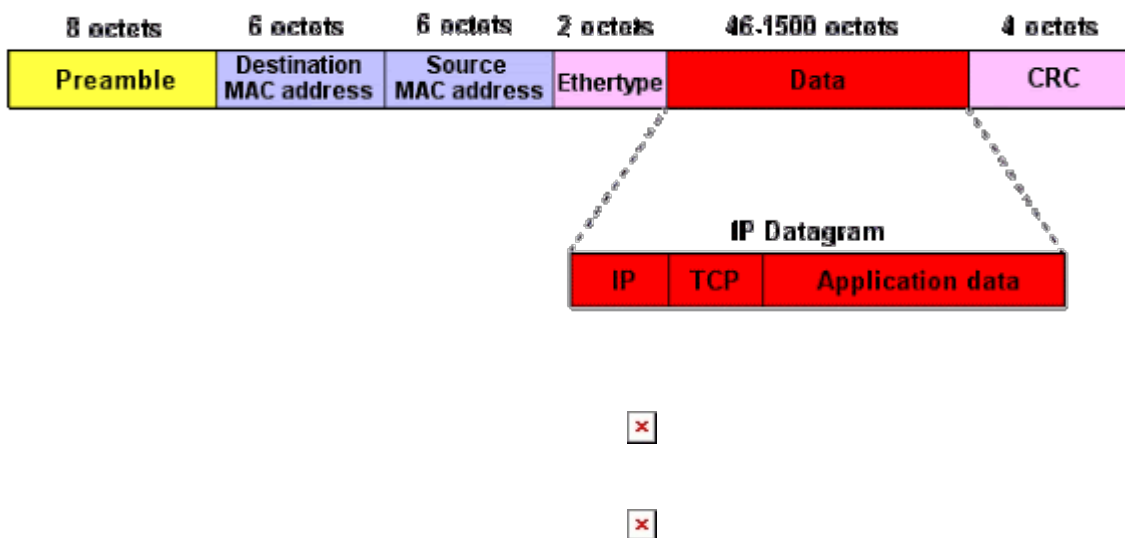
Ethernet basiert auf der Idee, dass die Teilnehmer eines LANs Nachrichten durch Hochfrequenz übertragen, allerdings **nur innerhalb eines gemeinsamen Leitungsnetzes**. Jede Netzwerkschnittstelle hat **einen global eindeutigen 48-Bit-Schlüssel**, der als **MAC-Adresse** (=Media-Access-Control-Adress) bezeichnet wird. Das stellt sicher, dass alle Systeme in einem Ethernet unterschiedliche Adressen haben.

Ethernet Frame

Bei der Übertragung von Daten über Ethernet ist das Ethernet-Frame hauptverantwortlich für die korrekte Regelsetzung und erfolgreiche Übermittlung von Datenpaketen. Versendete Daten über Ethernet werden vom Frame sozusagen getragen. Ein Ethernet-Frame ist zwischen 64 Byte und 1518 Byte groß, abhängig von der Größe der zu transportierenden Daten.

Im OSI-Modell befindet sich der Frame auf der Sicherungsschicht, die für die fehlerfreie Übertragung verantwortlich ist und trennt den Bitdatenstrom in Blöcke bzw. Frames auf.





Ein Ethernet-Frame muss standardmäßig mindestens 64 Byte groß sein, damit die Kollisionserkennung funktioniert, und kann maximal 1.518 Byte groß sein. Das Paket beginnt immer mit einer Präambel, die die Synchronisation zwischen Sender und Empfänger regelt und einem „Start Frame Delimiter“ (SFD), der das Frame definiert. Beide Informationen sind eine Bitfolge im Format 10101010... Im eigentlichen Frame finden sich Informationen zu Ziel- und Quelladressen (MAC-Format) und Steuerinformationen (im Fall von Ethernet II das Type-Field, später eine Längenangabe), dann folgt der zu übermittelnde Datensatz. Eine „Frame Check Sequence“ (FCS) schließt als Prüfsumme das gesamte Frame (ausgenommen Präambel und SFD). Das Paket wird von einem „Inter Frame Gap“ abgeschlossen, der eine 9,6 µs lange Sendepause festlegt.

Ethernet II benutzt die klassische Framestruktur, die das sogenannte Type-Field („Typ“) beinhaltet, womit verschiedene Protokolle der Vermittlungsschicht definiert werden. Im OSI-Modell ist die Vermittlungsschicht (auch „Network Layer“) wichtig für die Schaltung von Verbindungen und die Bereitstellung von Netzwerkadressen. Das Type-Field wurde in späteren Frame-Formaten durch eine Längenangabe ersetzt.

CSMA/CD

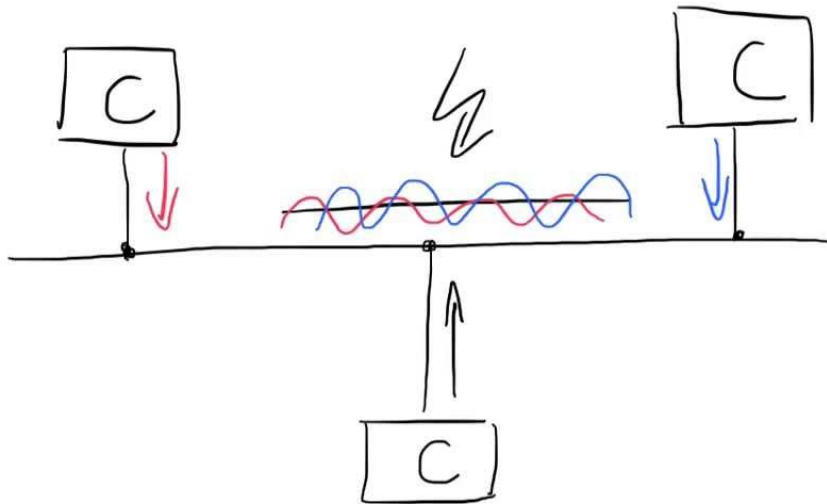
CSMA/CD (Carrier Sense Multiple Access with Collision Detection) ist das Zugriffsverfahren des Ethernets. Die Grundidee dabei ist, dass jede Station zu senden beginnen kann, wann sie will. Die einzelnen Stationen haben jederzeit und konkurrierend Zugang (Multiple Access) zum gemeinsamen Übertragungsmedium. Das grundlegende Motto könnte damit lauten:

Jeder darf, wann er will

Eingesetzt wird dieses Verfahren bei logischen Bus-Topologien. Dabei ist egal, ob physikalisch eine Bus- oder eine Stern-Topologie vorliegt.

Vorgehen

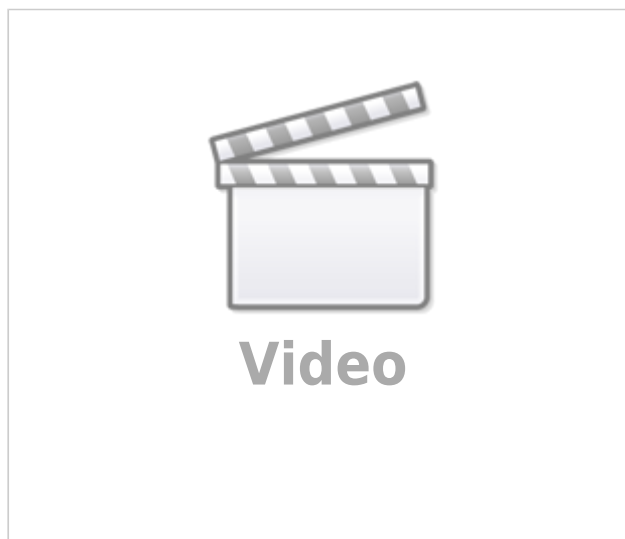
Durch Regelungen wird versucht, das Risiko zu minimieren, dass zwei Stationen ungünstigerweise gleichzeitig zu senden beginnen und somit die Signale auf dem Übertragungsweg zerstört/gestört werden (**=Kollision**).



1) Leitung prüfen (Carrier Sense)

Der erste Teil der Abkürzung steht für Kollisionsverhinderung. Dabei wird vor einer geplanten Sendung das Übertragungsmedium abgehört, ob dieses frei ist. Wenn das Medium frei ist, wird gesendet. -> **LISTEN BEFORE TALKING**

2) Erkennen von Kollisionen (Collision Detection) Kommt es trotzdem zu einer Kollision, weil zwei Stationen gleichzeitig zu senden beginnen (**Multiple Access**), dann muss diese Kollision erkannt (**Collision Detection**) und reagiert werden. Dabei müssen alle Stationen immer am Medium horchen, ob eine Kollision auftritt. Ist dies der Fall, so sendet die erste Station, die eine Kollision erkennt, ein sogenanntes JAM-Signal aus. Jede Station, die das JAM-Signal registriert, stoppt unmittelbar das Senden von Daten. Nach einer zufälligen Zeitspanne, wird das Medium wieder überprüft und anschließend wieder begonnen zu senden.



Frage: Wie kann bei einer physikalischen Stern-Topologie eine Kollision auftreten?

Vorteile

- Jeder kann zu jeder Zeit senden

Nachteile

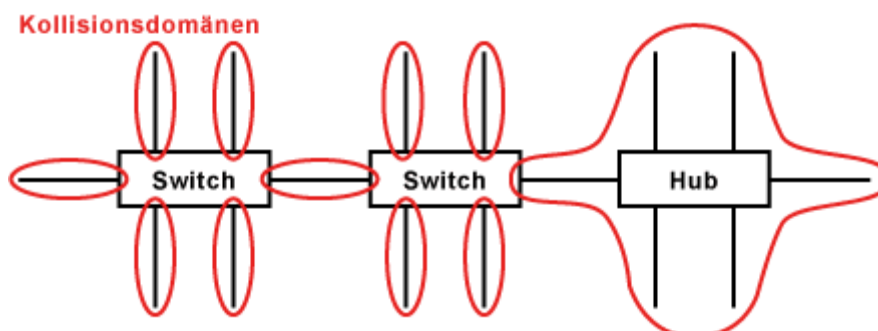
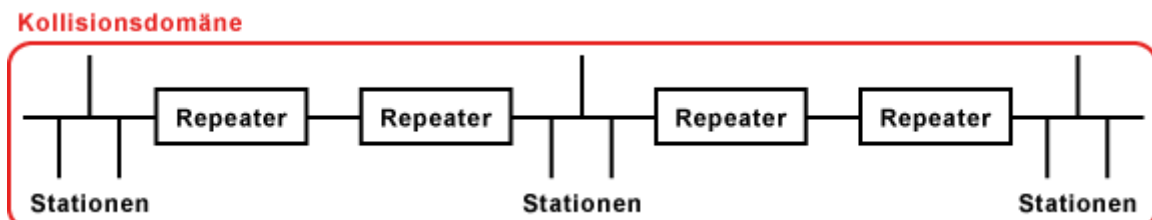
- Je mehr Stationen in einer Kollisionsdomäne, desto häufiger treten Kollisionen auf
- Der Zeitpunkt einer Sendung ist zufällig
- Das Verfahren ist ungeeignet für zeitkritischen Anwendungen

Kollisionsdomäne

Mit dem Begriff Kollisionsdomäne wird in einem Computernetz ein Teilbereich aus Teilnehmerstationen in derselben OSI-Modell-Schicht 1 bezeichnet. Eine Kollisionsdomäne umfasst alle Netzwerkgeräte, die um den Zugriff auf ein gemeinsames Übertragungsmedium konkurrieren. Das Übertragungsmedium ist daher eine zwischen allen Netzstationen geteilte Ressource. Grundlegende Vorstellung dabei ist, dass alle Netzwerkteilnehmer die Chance zur gleichberechtigten Nutzung des Netzwerkes besitzen.

Bei einem gemeinsamen Medium kann zu einer bestimmten Zeit nur jeweils eine Station Informationen übertragen, die an alle anderen Stationen übertragen bzw. von diesen empfangen wird. Fangen in einem derartigen gemeinsamen Schicht-1-Segment zwei Stationen gleichzeitig an zu senden, kommt es zu Kollisionen.

Beispiele für Kollisionsdomänen:



From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

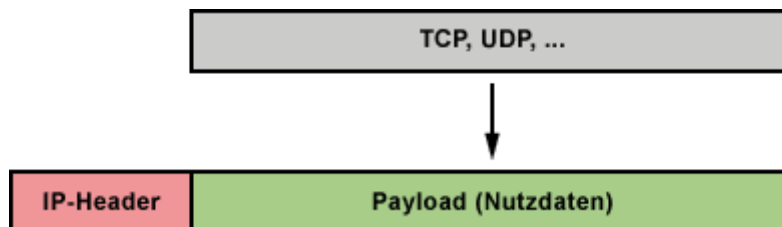
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4:4_09:4_09_00



Last update: **2020/03/17 07:54**

IPv4 - Internet Protocol Version 4 (Layer 3)

Das Internet Protocol, kurz IP, wird im Rahmen der Protokollfamilie TCP/IP zur Vermittlung von Datenpaketen verwendet. Es arbeitet auf der **Schicht 3 des OSI-Schichtenmodells** und hat maßgeblich die Aufgabe, **Datenpakete zu adressieren** und in einem dezentralen, **verbindungslosen und paketerorientierten Netzwerk zu übertragen**. Dazu haben alle Netzwerk-Teilnehmer eine eigene IP-Adresse im Netzwerk. Sie dient nicht nur zur Identifikation eines Hosts, sondern auch des Netzes, in dem sich der jeweilige Host befindet.

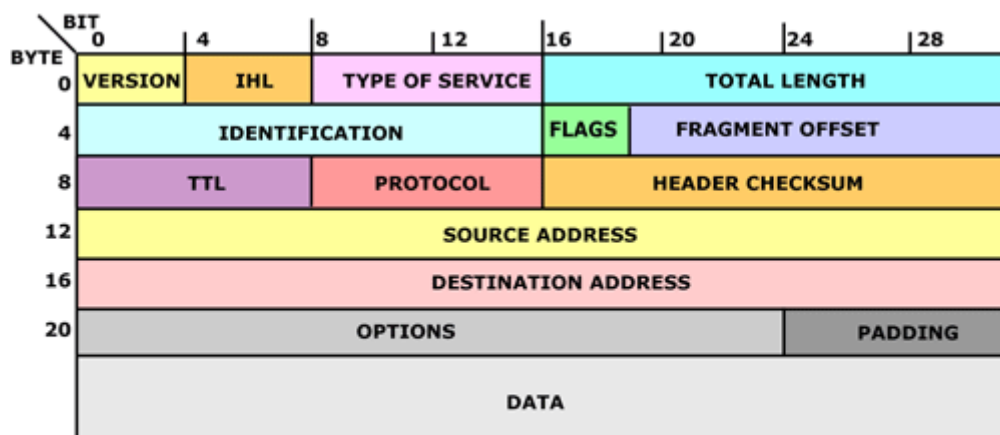


Aufgaben und Funktionen von IPv4

- Logische Adressierung (IPv4-Adresse)
- IPv4-Konfiguration
- IPv4-Header
- IP-Routing
- Namensauflösung (DNS-Dienst)

IPv4 Header

Jedes IPv4-Datenpaket besteht aus einem Header (Kopf) und dem Payload, in dem sich die Nutzdaten befinden. Der Header ist den Nutzdaten vorangestellt. Im IP-Header sind Informationen enthalten, die für die Verarbeitung durch das Internet Protocol notwendig sind.



Feldinhalt	Bit	Beschreibung
Version	4	Hier ist die Version des IP-Protokolls abgelegt, nach der das IP-Paket erstellt wurde.

Feldinhalt	Bit	Beschreibung
IHL	4	IHL = Internet Header Length gibt die Länge des IP-Headers als Vielfaches von 32 Bit an. Der Maximalwert von Binär 1111 (15) entspricht einer Header-Länge von $15 \times 32 \text{ Bit} = 480 \text{ Bit} = 60 \text{ Byte}$.
ToS	8	ToS = Type of Service legt die Qualität des angeforderten Dienstes fest. Das Feld unterteilt sich in Priorität (Priority) von 3 Bit und Eigenschaften für die Übertragung von 5 Bit.
Paketlänge (Total Length)	16	Enthält die Gesamtlänge des IP-Paketes. Abzüglich des IHL ergibt sich die Länge der reinen Nutzdaten.
Kennung(Identification)	16	Der Wert wird zur Nummerierung der Datenpakete verwendet. Die Kennung ist eindeutig und fortlaufend.
Flags	3	Da die Nutzdaten in der Regel nicht in ein IP-Paket hineinpassen, werden die Daten zerlegt und in mehrere IP-Pakete verpackt und verschickt. Man spricht dann von Fragmentierung. Die Flags gehen näher darauf ein. Das erste Flag ist immer 0. Das zweite Flag (DF) verbietet die Fragmentierung des Datenpakets, wenn es gesetzt ist. Das dritte Flag (MF) gibt weitere Datenpaket-Fragmente an, wenn es gesetzt ist.
Fragment-Offset	13	Enthält ein IP-Paket fragmentierte Nutzdaten, steht in diesem Feld die Position der Daten im ursprünglichen IP-Paket.
TTL (Time-to-Live)	8	Mit TTL gibt der Sender die Lebensdauer des Pakets in Sekunden an. Jede Station, die ein IP-Paket weiterleiten muss, zieht von diesem Wert mindestens 1 ab. Hat der TTL-Wert 0 erreicht, wird das IP-Paket verworfen. Dieser Mechanismus verhindert, dass Pakete ewig leben, wenn sie nicht zustellbar sind. TTL-Werte zwischen 30 und 64 sind typisch.
Protokoll	8	Dieses Feld enthält den Port des übergeordneten Transport-Protokolls (z. B. TCP oder UDP).
Header Checksumme	16	Diese Checksumme sichert die Korrektheit des IP-Headers. Für die Nutzdaten muss ein übergeordnetes Protokoll die Fehlerkorrektur übernehmen. Da sich die einzelnen Felder des IP-Headers ständig ändern, muss jede Station auf dem Weg zum Ziel die Checksumme prüfen und auch wieder neu berechnen. Um die Verzögerung gering zu halten wird deshalb nur der IP-Header des Paketes geprüft.
Quell-IP-Adresse (Source IP-Address)	32	An dieser Stelle steht die IP-Adresse der Station, die das IP-Paket abgeschickt hat (Sender).
Ziel-IP-Adresse (Destination IP-Address)	32	An dieser Stelle steht die IP-Adresse der Station, für die das IP-Paket bestimmt ist. Soll das IP-Paket an mehrere Stationen zugestellt werden, muss hier eine Multicast-Adresse stehen.
Optionen/Füllbits (Options/Padding)	32	Das Optionsfeld des IP-Headers enthält hauptsächlich Informationen zu Routing-, Debugging-, Statistik- und Sicherheitsfunktionen. Dieses Feld ist optional und kann bis zu 40 Byte lang sein. Es ist immer in 32 Bit aufgeteilt und wird bei Bedarf mit Nullen aufgefüllt. Auf die genauen Funktionen dieses Feldes wird hier nicht weiter eingegangen. Nur soviel sei noch gesagt: das Optionsfeld wird meist zu Diagnosezwecken verwendet.

Der Header ist in jeweils 32-Bit-Blöcke unterteilt. Dort sind Angaben zu Servicetypen, Paketlänge, Sender- und Empfängeradresse abgelegt. Ein IP-Paket muss mindestens 20 Byte Header und 8 Byte Nutzdaten bzw. Nutz- und Fülldaten enthalten. Die Gesamtlänge eines IP-Paketes darf 65.535 Byte nicht überschreiten.

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4:4_09:4_09_01

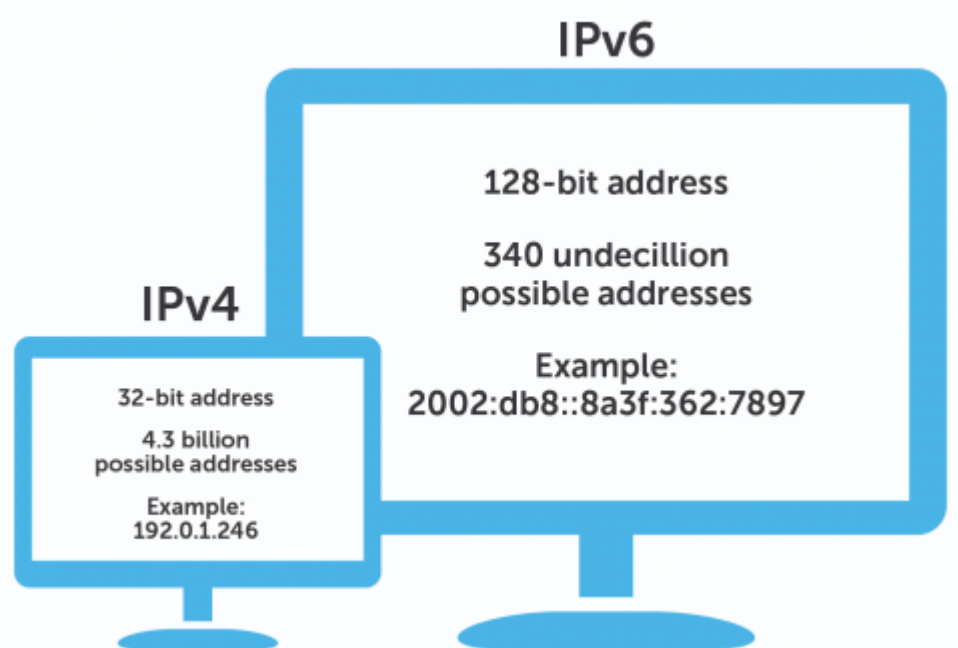


Last update: **2020/03/17 21:07**

IPv6 - Internet Protocol Version 6 (Layer 3)

IPv6 ist als Internet Protocol (Version 6) für die Vermittlung von Datenpaketen durch ein paketvermittelndes Netz, die Adressierung von Netzknoten und -stationen, sowie die Weiterleitung von Datenpaketen zwischen Teilnetzen zuständig. Mit diesen Aufgaben ist IPv6 der Schicht 3 des OSI-Schichtenmodells zugeordnet. Die Aufgabe des Internet-Protokolls besteht im Wesentlichen darin, Datenpakete von einem System über verschiedene Netzwerke hinweg zu einem anderen System zu vermitteln (Routing).

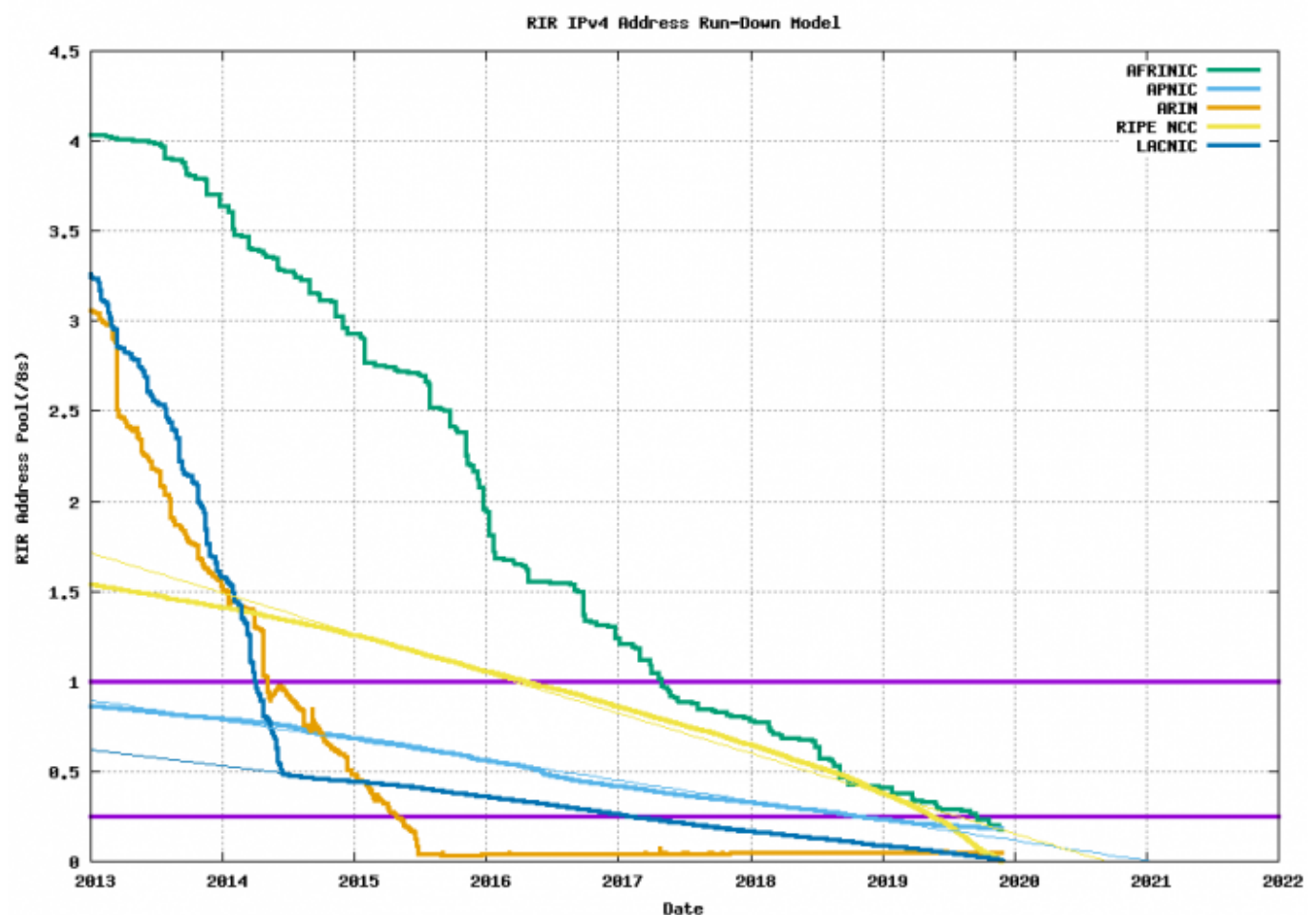
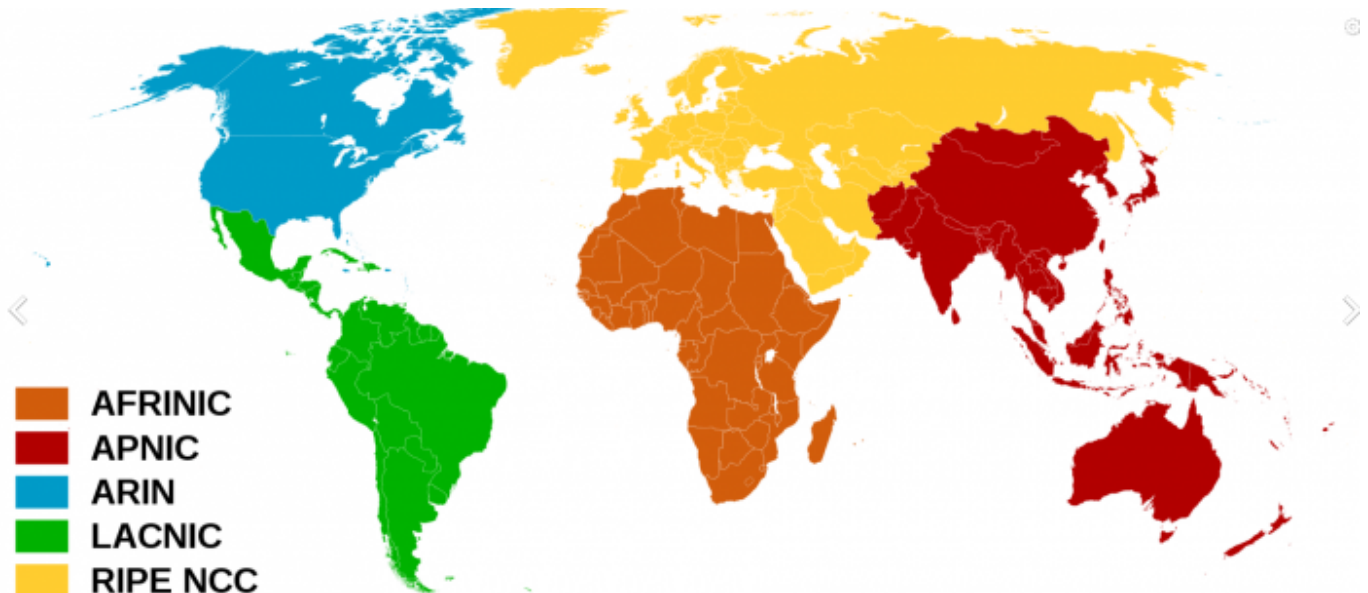
IPv6 ist der direkte Nachfolger von IPv4 und Teil der Protokollfamilie TCP/IP. Seit Dezember 1998 steht IPv6 bereit und wurde hauptsächlich wegen der Adressknappheit und verschiedener Unzulänglichkeiten von IPv4 entwickelt spezifiziert. Da weltweit immer mehr Menschen, Maschinen und Geräte an das Internet mit einer eindeutigen Adresse angeschlossen werden sollen, reichen die 4 Milliarden IPv4-Adressen nicht mehr aus.



Warum IPv6?

IPv6 gilt als Wunderwaffe gegen so manche Probleme mit Netzwerkprotokollen und gleichzeitig wird es als Teufelszeug verdammt, das wieder neue unbekannte Probleme hervorruft. Eine Tatsache ist, dass Administratoren, Programmierer und Hersteller IPv6 neu lernen müssen. Viele Rezepte aus der IPv4-Welt taugen unter IPv6 nicht mehr. Erschwerend kommt hinzu, dass es bei IPv6 allen Beteiligten an Erfahrung fehlt. IPv6-Gurus, die man bei einem großen Problem befragen kann, gibt es nicht so viele.

Bei IPv6 ist das Ende-zu-Ende-Prinzip konsequent weiter gedacht. Ein Interface kann mehrere IPv6-Adressen haben und es gibt spezielle IPv6-Adressen, denen mehrere Interfaces zugeordnet sind. IPv6 löst also nicht nur die Adressknappheit, sondern bietet auch Erleichterungen bei der Konfiguration und im Betrieb. Die zustandslose IPv6-Konfiguration und verbindungslokalen Adressen, die bereits nach dem Computerstart verfügbar sind, vereinfachen die Einrichtung und den Betrieb eines lokalen Netzwerks. Damit das gelingt sind Planer und Errichter von IP-Netzen gefordert sich eine neue Denkweise anzueignen.



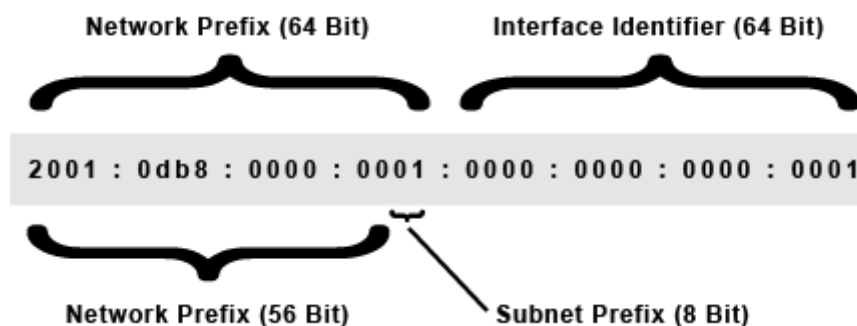
[das_war_s_mit_ipv4-adressen_in_europa_heise_online.pdf](#)

IPv6 Adressen

Eine IPv6-Adresse ist eine Netzwerk-Adresse, die einen Host eindeutig innerhalb eines IPv6-Netzwerks logisch adressiert. Die Adresse wird auf IP- bzw. Vermittlungsebene (des OSI-Schichtenmodells) benötigt, um Datenpakete verschicken und zustellen zu können. Im Gegensatz zu anderen Adressen hat ein IPv6-Host mehrere IPv6-Adressen, die unterschiedliche Gültigkeitsbereiche haben. Konkret

bedeutet das, dass wenn von IPv6-Adressen die Rede ist, dass nicht immer klar ist, welchen Gültigkeitsbereich diese IPv6-Adressen aufweisen. Grob unterscheidet man zwischen verbindungslokalen und globalen IPv6-Adressen. Die verbindungslokale IPv6-Adresse ist nur im lokalen Netzwerk gültig und wird nicht geroutet. Die globale IPv6-Adresse ist über das lokale Netzwerk hinaus im Internet gültig.

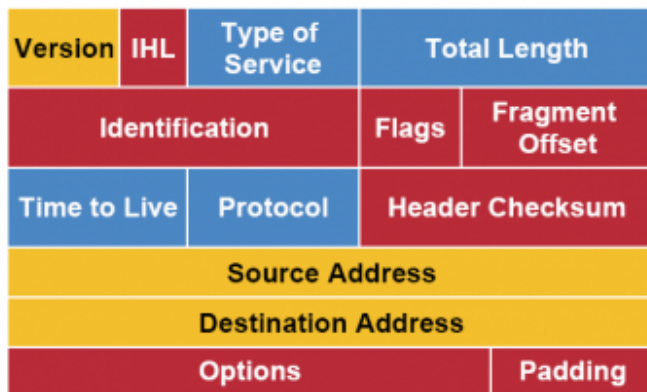
Eine IPv6-Adresse hat eine Länge von 128 Bit. Diese Adresslänge erlaubt eine unvorstellbare Menge von 2¹²⁸ oder 3,4 x 10³⁸ IPv6-Adressen. Das sind 340.282.366.900.000.000.000.000.000.000.000.000.000 IPv6-Adressen, also rund 340 Sextillionen Adressen. Bei IPv4 spricht man von rund 4,3 Milliarden Adressen. Der Adressraum von IPv6 reicht aus, um umgerechnet jeden Quadratmillimeter der Erdoberfläche inklusive der Ozeane mit rund 600 Billionen Adressen zu pflastern.



Eine IPv6-Adresse besteht aus 128 Bit. Wegen der unhandlichen Länge werden die 128 Bit in 8 mal 16 Bit unterteilt. Je 4 Bit werden als eine hexadezimale Zahl dargestellt. Je 4 Hexzahlen werden gruppiert und durch einen Doppelpunkt („:“) getrennt. Um die Schreibweise zu vereinfachen lässt man führende Nullen in den Blöcken weg. Eine Folge von 8 Nullen kann man durch zwei Doppelpunkte („::“) ersetzen.

Eine IPv6-Adresse besteht aus zwei Teilen. Dem Network Prefix (Präfix oder Netz-ID) und dem Interface Identifier (Suffix, IID oder EUI). Der Network Prefix kennzeichnet das Netz, Subnetz bzw. Adressbereich. Der Interface Identifier kennzeichnet einen Host in diesem Netz. Er wird aus der 48-Bit-MAC-Adresse des Interfaces gebildet und dabei in eine 64-Bit-Adresse umgewandelt. Es handelt sich dabei um das Modified-EUI-64-Format. Auf diese Weise ist das Interface unabhängig vom Network Prefix eindeutig identifizierbar.

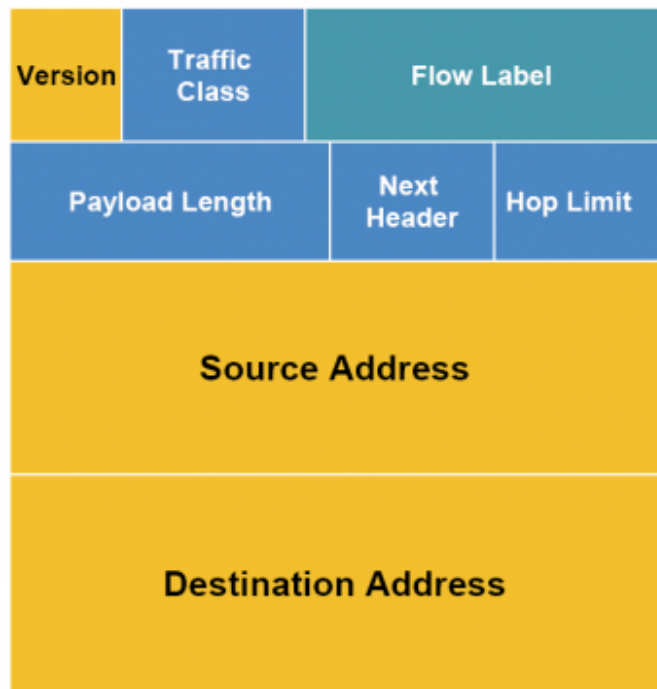
IPv4 Header



Legend

- Field's Name Kept from IPv4 to IPv6
- Fields Not Kept in IPv6
- Name and Position Changed in IPv6
- New Field in IPv6

IPv6 Header



ipv6_-_neues_internet-protokoll.pdf

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4:4_09:4_09_02



Last update: **2020/03/17 08:59**

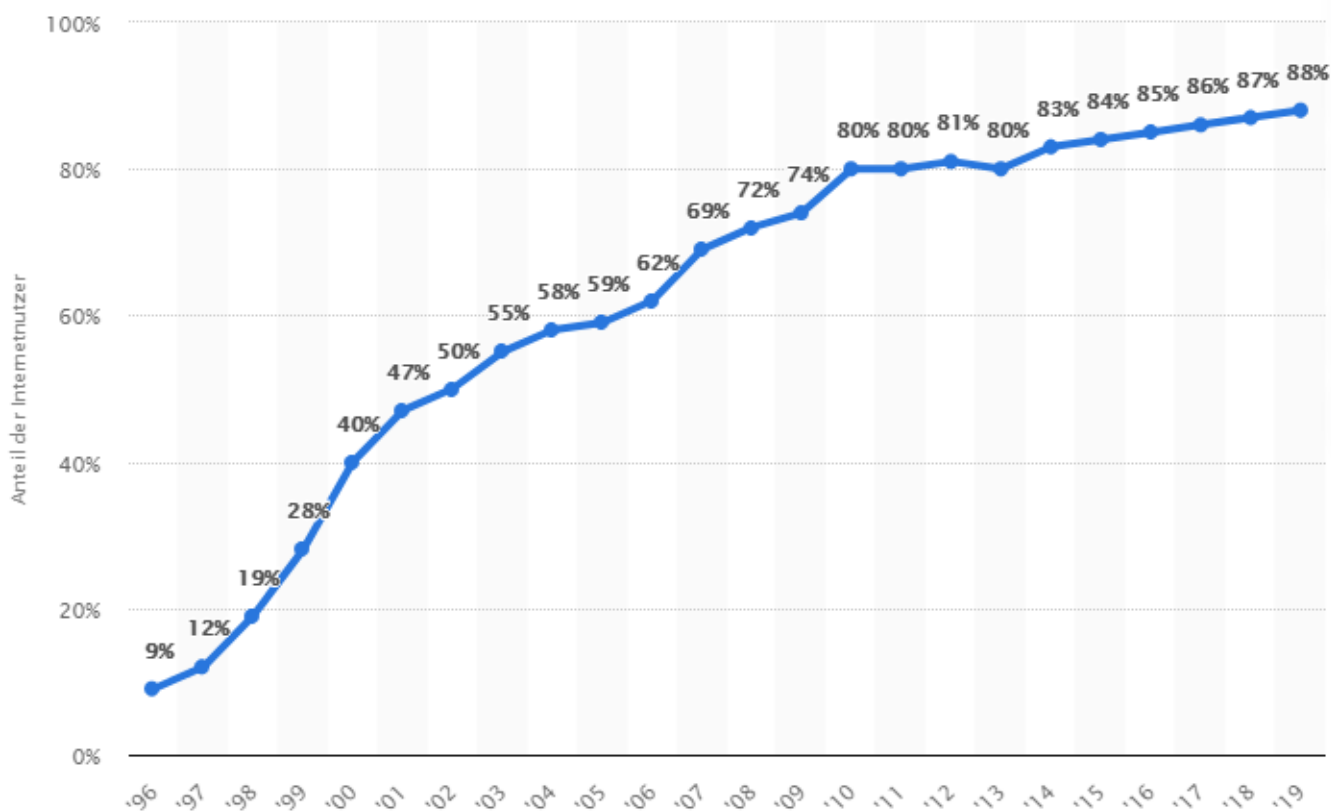
NAT - Network Address Translation (Layer 3)

NAT (Network Address Translation) ist ein Verfahren, dass in **IP-Routern eingesetzt** wird, die lokale Netzwerke mit dem Internet verbinden. Weil Internet-Zugänge in der Regel nur über **eine einzige öffentliche** und damit routbare IPv4-Adresse verfügen, müssen sich alle anderen Hosts im lokalen Netzwerk mit privaten IPv4-Adressen begnügen. **Private IP-Adressen** dürfen zwar **mehrfach verwendet** werden, aber besitzen **in öffentlichen Netzen keine Gültigkeit**. Hosts mit einer privaten IPv4-Adresse können somit nicht mit Hosts außerhalb des lokalen Netzwerks kommunizieren.

NAT ist allerdings nur eine mittlerweile sehr lang andauernde Notlösung, um die Adressknappheit von IPv4 zu umgehen. Um die damit einhergehenden Probleme zu lösen muss langfristig auf ein Internet-Protokoll mit einem größeren Adressraum umgestellt werden. IPv6 ist ein solches Protokoll.

Warum NAT?

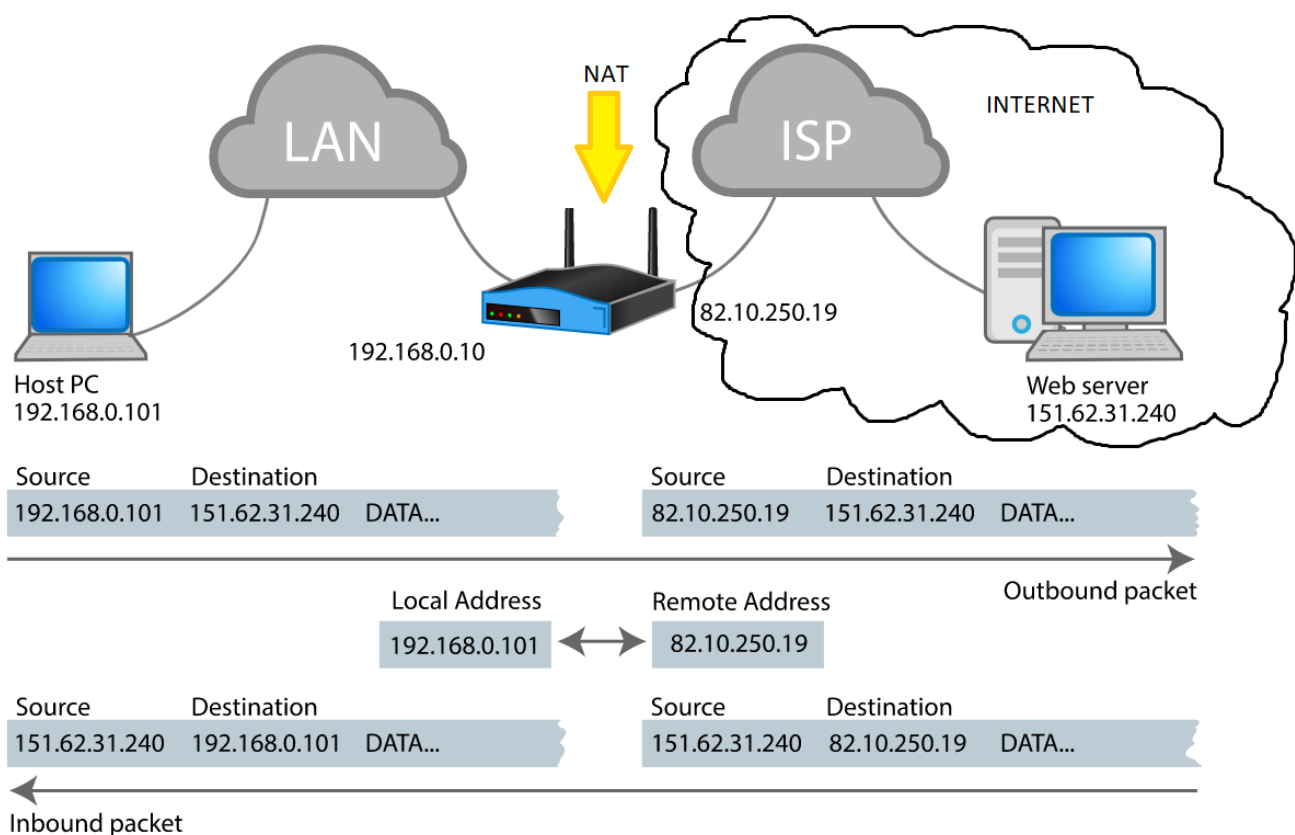
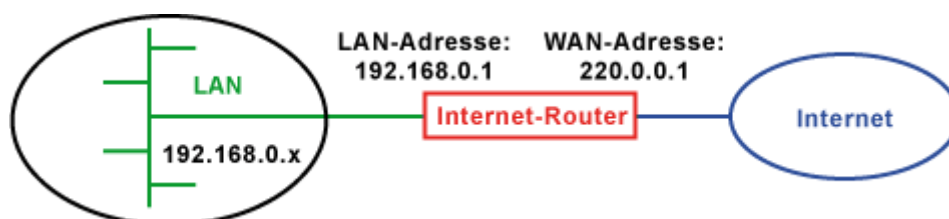
Die **ersten IPv4-Netze** waren **anfangs** eigenständige Netz **ohne Verbindung nach außen**. Hier begnügte man sich mit IPv4-Adressen aus den privaten Adressbereichen. Parallel dazu kam es bereits **Ende der 1990er** Jahre zu **Engpässen bei öffentlichen IPv4-Adressen**. Die steigende Anzahl der Einwahlzugänge über das Telefonnetz mussten mit IPv4-Adressen versorgt werden. Bis heute bekommt **ein Internet-Anschluss** nur eine **IPv4-Adresse für ein Gerät**. Damals war es undenkbar, dass an einem Internet-Anschluss ein ganzes Heimnetzwerk betrieben wird. Wenn ein Haushalt einen PC per Modem an das Telefonnetz angeschlossen und sich ins Internet eingewählt hat, dann war das schon etwas besonderes.

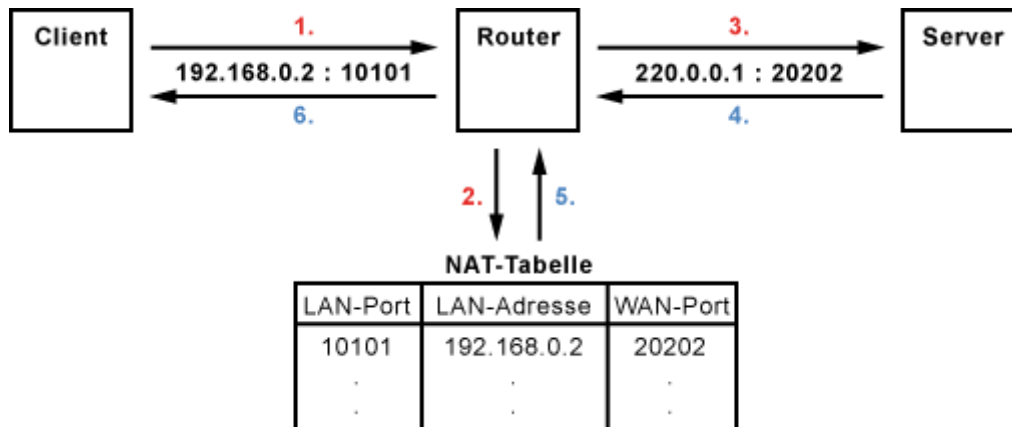


das_war_s_mit_ipv4-adressen_in_europa_heise_online.pdf

NAT

Der Betrieb eines NAT-Routers ist üblicherweise an einem gewöhnlichen Internet-Anschluss. Zum Beispiel über DSL oder Kabelmodem. Der eingesetzte Router dient als Zugang zum Internet und als Standard-Gateway für das lokale Netzwerk. In der Regel wollen über den Router mehr Geräte ins Internet, als öffentliche IP-Adressen zur Verfügung stehen. In der Regel nur eine einzige. Beispielsweise bekommt der Router des lokalen Netzwerks die öffentliche IP-Adresse 222.0.0.1 für seinen WAN-Port vom Internet Service Provider (ISP) zugewiesen. Weil nur eine öffentliche IP-Adresse vom Internet-Provider zugeteilt wurde, bekommen die Stationen im LAN private IP-Adressen aus speziell dafür reservierten Adressbereichen zugewiesen. Diese Adressen sind nur innerhalb des privaten Netzwerks gültig. Private IP-Adressen werden in öffentlichen Netzen nicht geroutet. Das bedeutet, dass Stationen mit privaten IP-Adressen keine Verbindung ins Internet bekommen können. Damit das trotzdem funktioniert, wurde NAT entwickelt.

**Ablauf von NAT**



1. Der Client schickt seine Datenpakete mit der IP-Adresse 192.168.0.2 und dem TCP-Port 10101 an sein Standard-Gateway, bei dem es sich um einen NAT-Router handelt.
2. Der NAT-Router tauscht IP-Adresse (LAN-Adresse) und TCP-Port (LAN-Port) aus und speichert beides mit der getauschten Port-Nummer (WAN-Port) in der NAT-Tabelle.
3. Der Router leitet das Datenpaket mit der WAN-Adresse 220.0.0.1 und der neuen TCP-Port 20202 ins Internet weiter.
4. Der Empfänger (Server) verarbeitet das Datenpaket und schickt seine Antwort zurück.
5. Der NAT-Router stellt nun anhand der Port-Nummer 20202 (WAN-Port) fest, für welche IP-Adresse (LAN-Adresse) das Paket im lokalen Netz gedacht ist.
6. Er tauscht die IP-Adresse und die Port-Nummer wieder aus und leitet das Datenpaket ins lokale Netz weiter, wo es der Client entgegen nimmt.

Probleme

- Durch NAT können nur noch die, die über öffentliche IPv4-Adressen und in der Regel auch über das notwendige Kleingeld verfügen, Dienste im Internet anbieten.
- Die Einträge in der NAT-Tabelle des Routers sind nur für eine kurze Zeit gültig. Für eine Anwendung, die nur sehr unregelmäßig Daten austauscht, bedeutet das, dass ständig die Verbindung abgebrochen wird und dadurch die Erreichbarkeit eingeschränkt ist.

Vorteile

- Mehr Sicherheit - PCs im Netzwerk nicht direkt ansprechbar = mehr Privatsphäre
- Firewall - keine von außen eingehenden Verbindungen erlaubt, so fern keine Verbindung von intern aufgebaut wurde

From:
<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4:4_09:4_09_03



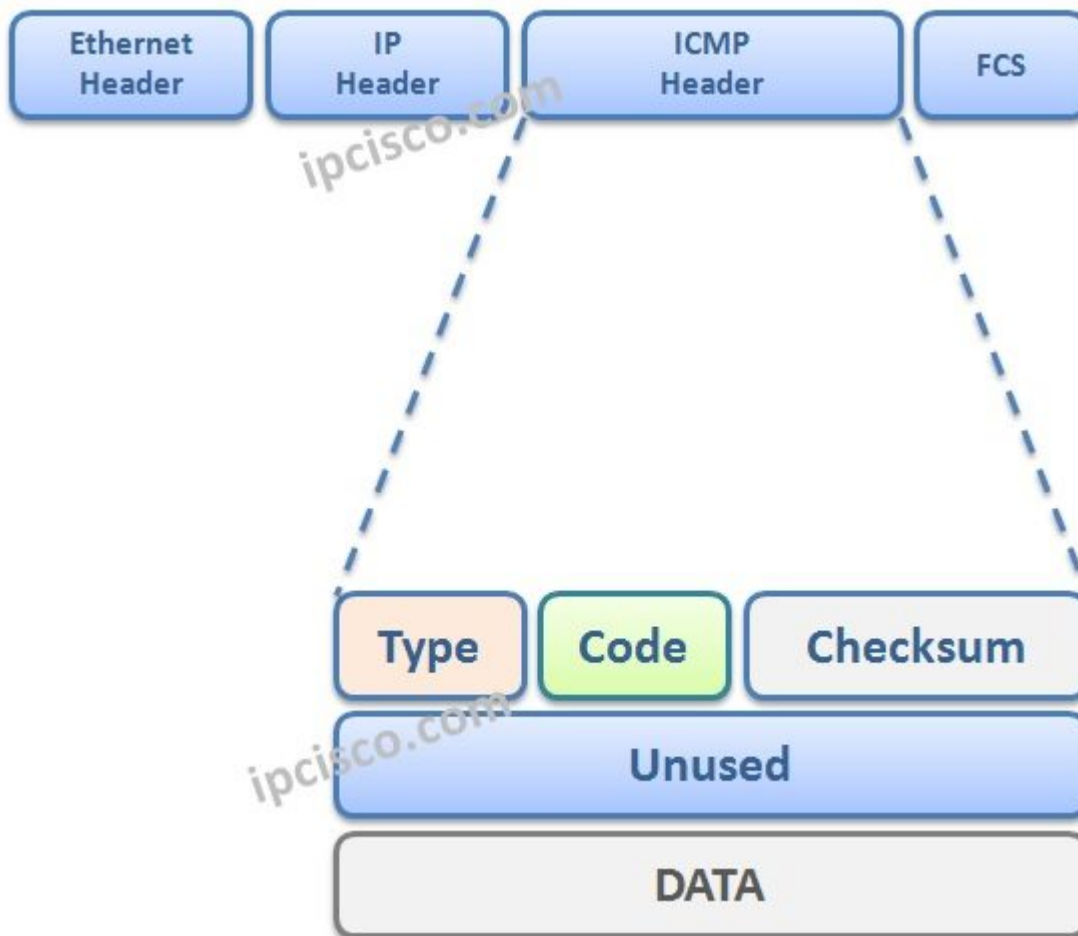
Last update: **2020/03/18 08:18**

ICMP - Internet Control Message Protocol (Layer 3)

Das Internet Control Message Protocol (ICMP) ist Bestandteil des Internet Protocols (IP). Es wird aber als eigenständiges Protokoll behandelt, das zur Übermittlung von Meldungen über IP dient.

Hauptaufgabe von ICMP ist die **Übertragung von Statusinformationen und Fehlermeldungen der Protokolle IP, TCP und UDP**. Die ICMP-Meldungen werden zwischen Rechnern und aktiven Netzknoten, z. B. Routern, benutzt, um sich gegenseitig Probleme mit Datenpaketen mitzuteilen. Ziel ist, die Übertragungsqualität zu verbessern.

Jedes Betriebssystem mit TCP/IP hat Tools, die ICMP nutzen. Zwei bekannte Tools sind Ping und Trace Route. Beides sind sehr einfache Programme, die zur Analyse von Netzwerk-Problemen gedacht sind und damit wesentlich zur Problemlösung beitragen können.



Typ	Code	Bedeutung
0	0	Ping, Echo Antwort
3	0	Netzwerk nicht erreichbar
3	1	Host nicht erreichbar
3	2	Ziel-Protokoll nicht verfügbar
3	3	Ziel-Port nicht erreichbar
3	6	Netzwerk unbekannt
3	7	Ziel-Host unbekannt
4	0	Überlastkontrolle, ungenutzt
8	0	Ping, Echo-Anfrage
9	0	Routen-Bekanntmachung
10	0	Router-Discovery
11	0	Trace-Route
12	0	Schlechter IP-Header

Aufzeichnung mit Wireshark

```
C:\Users\Andy>ping www.orf.at

Ping wird ausgeführt für www.orf.at [194.232.104.141] mit 32 Bytes Daten:
Antwort von 194.232.104.141: Bytes=32 Zeit=11ms TTL=55
Antwort von 194.232.104.141: Bytes=32 Zeit=12ms TTL=55
Antwort von 194.232.104.141: Bytes=32 Zeit=11ms TTL=55
Antwort von 194.232.104.141: Bytes=32 Zeit=12ms TTL=55

Ping-Statistik für 194.232.104.141:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0
    (0% Verlust),
Ca. Zeitangaben in Millisek.:
    Minimum = 11ms, Maximum = 12ms, Mittelwert = 11ms
```


1.
2.
3.
4.

No.	Time	Source	Destination	Protocol	Length	Info
205	8.472486	192.168.1.29	194.232.104.141	ICMP	74	Echo (ping) request id=0x0001, seq=302/11777, ttl=128 (reply in 212)
212	8.484322	194.232.104.141	192.168.1.29	ICMP	74	Echo (ping) reply id=0x0001, seq=302/11777, ttl=55 (request in 205)
215	9.476717	192.168.1.29	194.232.104.141	ICMP	74	Echo (ping) request id=0x0001, seq=303/12033, ttl=128 (reply in 216)
216	9.488903	194.232.104.141	192.168.1.29	ICMP	74	Echo (ping) reply id=0x0001, seq=303/12033, ttl=55 (request in 215)
222	10.481188	192.168.1.29	194.232.104.141	ICMP	74	Echo (ping) request id=0x0001, seq=304/12289, ttl=128 (reply in 223)
223	10.492692	194.232.104.141	192.168.1.29	ICMP	74	Echo (ping) reply id=0x0001, seq=304/12289, ttl=55 (request in 222)
246	11.487172	192.168.1.29	194.232.104.141	ICMP	74	Echo (ping) request id=0x0001, seq=305/12545, ttl=128 (reply in 247)
247	11.499758	194.232.104.141	192.168.1.29	ICMP	74	Echo (ping) reply id=0x0001, seq=305/12545, ttl=55 (request in 246)

> Frame 246: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{428B0DD7-40A8-43CC-A1F0-BDC274827A6C}, id 0

> Ethernet II, Src: HewlettP_0b:2f:71 (24:be:05:0b:2f:71), Dst: Netgear_3e:59:a9 (b0:7f:b9:3e:59:a9)

> Internet Protocol Version 4, Src: 192.168.1.29, Dst: 194.232.104.141

Internet Control Message Protocol

Type: 8 (Echo (ping) request)

Code: 0

Checksum: 0x4c2a [correct]

[Checksum Status: Good]

Identifier (BE): 1 (0x0001)

Identifier (LE): 256 (0x0100)

Sequence number (BE): 305 (0x0131)

Sequence number (LE): 12545 (0x3101)

[Response frame: 247]

Data (32 bytes)

Data: 6162636465666768696a6b6c6d6e6f707172737475767761...

[Length: 32]

```

0000  b0 7f b9 3e 59 a9 24 be 05 0b 2f 71 08 00 45 00  ...>Y.$  ..-/q..E.
0010  00 3c 27 11 00 00 80 01 00 00 c0 a8 01 1d c2 e8  -<'.....
0020  68 8d 08 00 4c 2a 00 01 01 31 61 62 63 64 65 66  h...L*... 1abcdef
0030  67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76  ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67 68 69                    wabcdefg hi

```

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4:4_09:4_09_04

Last update: 2020/03/17 08:20

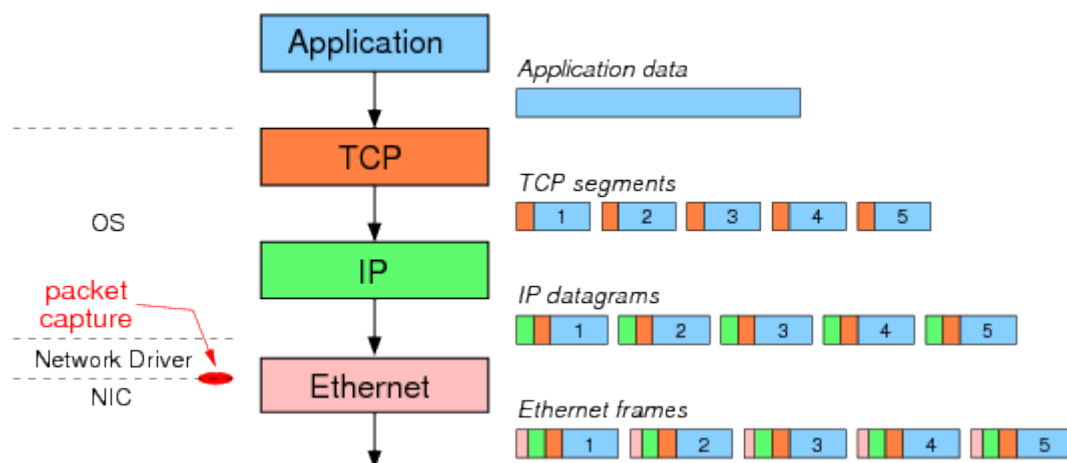
TCP - Transmission Control Protocol (Layer 4)

Das **Transmission Control Protocol** (=Übertragungssteuerungsprotokoll) ist ein Netzwerkprotokoll, das definiert, auf welche Art und Weise Daten zwischen Netzwerkkomponenten ausgetauscht werden sollen. Nahezu sämtliche aktuellen Betriebssysteme moderner Computer beherrschen TCP und nutzen es für den Datenaustausch mit anderen Rechnern. Das Protokoll ist ein zuverlässiges, verbindungsorientiertes, paketvermittelltes Transportprotokoll in Computernetzwerken. Es ist Teil der Internetprotokollfamilie, der Grundlage des Internets.

Aufgaben des TCP-Protokolls

Segmentierung (Data Segmenting)

Eine Funktion von TCP besteht darin, den von den Anwendungen kommenden **Datenstrom in Datenpakete bzw. Segmente aufzuteilen (Segmentierung)** und beim Empfang wieder zusammenzusetzen. Die Segmente werden mit einem **Header** versehen, in dem Steuer- und Kontroll-Informationen enthalten sind. Danach werden die **Segmente an das Internet Protocol (IP) übergeben**. Da beim IP-Routing die **Datenpakete unterschiedliche Wege** gehen können, entstehen unter Umständen **zeitliche Verzögerungen**, die dazu führen, dass die Datenpakete beim Empfänger in einer anderen Reihenfolge eingehen, als sie ursprünglich hatten. Deshalb werden die Segmente beim Empfänger auch wieder in die **richtige Reihenfolge** gebracht und erst dann an die adressierte Anwendung übergeben. Dazu werden die Segmente mit einer **fortlaufenden Sequenznummer** versehen (Sequenzierung).



Verbindungsmanagement (Connection Establishment and Termination)

Als **verbindungsorientiertes Protokoll** ist TCP für den **Verbindungsaufbau und Verbindungsabbau** zwischen zwei Stationen einer **Ende-zu-Ende-Kommunikation** zuständig. Durch TCP stehen Sender und Empfänger ständig in Kontakt zueinander.

Fehlerbehandlung (Error Detection)

Obwohl es sich eher um eine virtuelle Verbindung handelt, werden während der Datenübertragung

ständig **Kontrollmeldungen** ausgetauscht. Der Empfänger bestätigt dem Sender jedes empfangene Datenpaket. Trifft keine Bestätigung beim Absender ein, wird das Paket noch mal verschickt. Da es bei Übertragungsproblemen zu doppelten Datenpaketen und Quittierungen kommen kann, werden alle TCP-Pakete und TCP-Meldungen mit einer **fortlaufenden Sequenznummer** gekennzeichnet. So sind Sender und Empfänger in der Lage, die Reihenfolge und Zuordnung der Datenpakete und Meldungen zu erkennen.

Flusssteuerung (Flow Control)

Bei einer paketorientierten Übertragung ohne feste zeitliche Zuordnung und ohne Kenntnis des Übertragungswegs erhält das Transport-Protokoll vom Übertragungssystem **keine Information über die verfügbare Bandbreite**. Mit der **Flusssteuerung** werden beliebig langsame oder schnelle **Übertragungsstrecken dynamisch auszulasten** und auch auf unerwartete Engpässe und Verzögerungen reagiert.

Anwendungsunterstützung (Application Support)

TCP- und UDP-Ports sind eine Software-Abstraktion, um Kommunikationsverbindungen voneinander unterscheiden zu können. Ähnlich wie IP-Adressen Rechner in Netzwerken adressieren, **adressieren Ports spezifische Anwendungen** oder Verbindungen, die auf einem Rechner laufen.

Aufbau eines TCP Headers

Aufbau des TCP-Headers TCP-Pakete setzen sich aus dem Header-Bereich und dem Daten-Bereich zusammen. Im Header sind alle Informationen enthalten, die für eine gesicherte TCP-Verbindung wichtig sind. Der TCP-Header ist in mehrere 32-Bit-Blöcke aufgeteilt. Mindestens enthält der Header 5 solcher Blöcke. Somit hat ein TCP-Header eine Länge von **mindestens 20 Byte**.

Transmission Control Protocol (TCP) Header

20-60 bytes

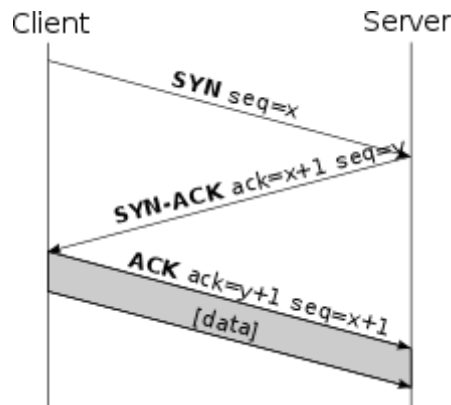
source port number 2 bytes				destination port number 2 bytes			
sequence number 4 bytes							
acknowledgement number 4 bytes							
data offset 4 bits		reserved 3 bits		control flags 9 bits		window size 2 bytes	
checksum 2 bytes				urgent pointer 2 bytes			
optional data 0-40 bytes							

TCP Kommunikation

Verbindungsaufbau - 3-Way-Handshake

Der Client, der eine Verbindung aufbauen will, sendet dem Server ein **SYN-Paket** (von englisch synchronize) mit einer Sequenznummer x. Die Sequenznummern sind dabei für die Sicherstellung einer vollständigen Übertragung in der richtigen Reihenfolge und ohne Duplikate wichtig. Es handelt sich also um ein Paket, dessen SYN-Bit im Paketkopf gesetzt ist (siehe TCP-Header). Die Start-Sequenznummer ist eine beliebige Zahl, deren Generierung von der jeweiligen TCP-Implementierung abhängig ist. Sie sollte jedoch möglichst zufällig sein, um Sicherheitsrisiken zu vermeiden.

Der Server (siehe Abbildung) empfängt das Paket. Ist der Port geschlossen, antwortet er mit einem TCP-RST, um zu signalisieren, dass keine Verbindung aufgebaut werden kann. Ist der Port geöffnet, bestätigt er den Erhalt des ersten SYN-Pakets und stimmt dem Verbindungsaufbau zu, indem er ein **SYN/ACK-Paket** zurückschickt (ACK von engl. acknowledgement ‚Bestätigung‘). Das gesetzte ACK-Flag im TCP-Header kennzeichnet diese Pakete, welche die Sequenznummer x+1 des SYN-Pakets im Header enthalten. Zusätzlich sendet er im Gegenzug seine Start-Sequenznummer y, die ebenfalls beliebig und unabhängig von der Start-Sequenznummer des Clients ist.



Der Client bestätigt zuletzt den Erhalt des SYN/ACK-Pakets durch das Senden eines eigenen **ACK-Pakets** mit der Sequenznummer $x+1$. Dieser Vorgang wird auch als **Forward Acknowledgement** bezeichnet. Aus Sicherheitsgründen sendet der Client den Wert $y+1$ (die Sequenznummer des Servers + 1) im ACK-Segment zurück. Die Verbindung ist damit aufgebaut. Im folgenden Beispiel wird der Vorgang abstrakt dargestellt:

1. SYN-SENT	→	<SEQ=100><CTL=SYN>	→	SYN-RECEIVED
2. SYN/ACK-RECEIVED	←	<SEQ=300><ACK=101><CTL=SYN,ACK>	←	
SYN/ACK-SENT				
3. ACK-SENT	→	<SEQ=101><ACK=301><CTL=ACK>	→	ESTABLISHED



TCP - 3 Wege Handshake

Einmal aufgebaut, ist die Verbindung für beide Kommunikationspartner gleichberechtigt, man kann einer bestehenden Verbindung auf TCP-Ebene nicht ansehen, wer der Server und wer der Client ist. Daher hat eine Unterscheidung dieser beiden Rollen in der weiteren Betrachtung keine Bedeutung mehr.

Aufzeichnung mit Wireshark

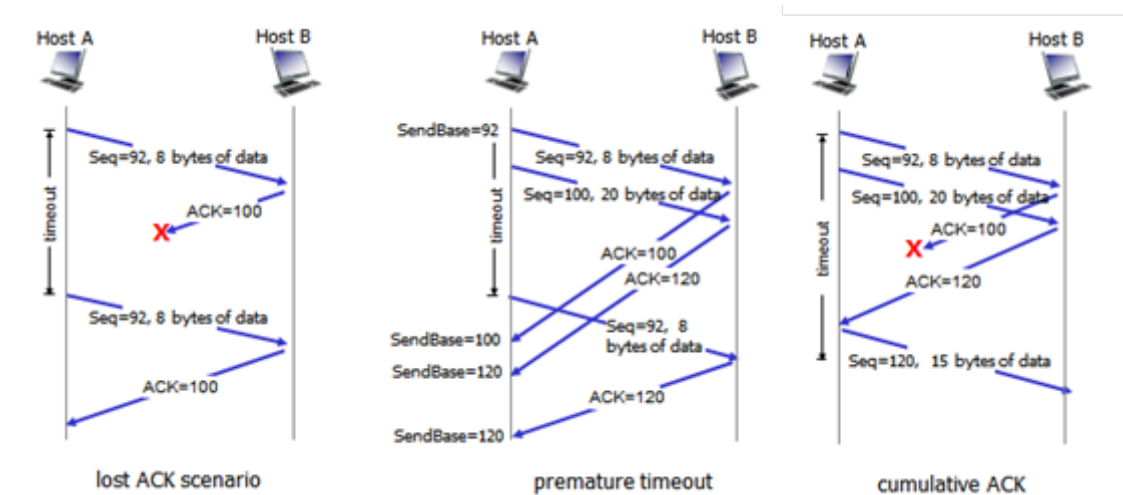
1072	25.068305	192.168.1.29	194.232.104.142	TCP	66	59726 → 80	[SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
1073	25.079954	194.232.104.142	192.168.1.29	TCP	66	80 → 59725	[SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1408 SACK_PERM=1 WS=128
1074	25.080052	192.168.1.29	194.232.104.142	TCP	54	59725 → 80	[ACK] Seq=1 Ack=1 Win=66048 Len=0

Frame 1073: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
 Ethernet II, Src: Netgear_3e:59:a9 (b0:7f:b9:3e:59:a9), Dst: HewlettP_0b:2f:71 (24:be:05:0b:2f:71)
 Internet Protocol Version 4, Src: 194.232.104.142, Dst: 192.168.1.29
 Transmission Control Protocol, Src Port: 80, Dst Port: 59725, Seq: 0, Ack: 1, Len: 0

Source Port: 80
 Destination Port: 59725
 [Stream index: 29]
 [TCP Segment Len: 0]
 Sequence number: 0 (relative sequence number)
 Acknowledgment number: 1 (relative ack number)
 1000 = Header Length: 32 bytes (8)
 > Flags: 0x012 (SYN, ACK)
 Window size value: 29200
 [Calculated window size: 29200]
 Checksum: 0x15a1 [unverified]
 [Checksum Status: Unverified]
 Urgent pointer: 0
 > Options: (12 bytes), Maximum segment size, No-Operation (NOP), No-Operation (NOP), SACK permitted, No-Operation (NOP), Window scale
 > [SEQ/ACK analysis]

Datenaustausch

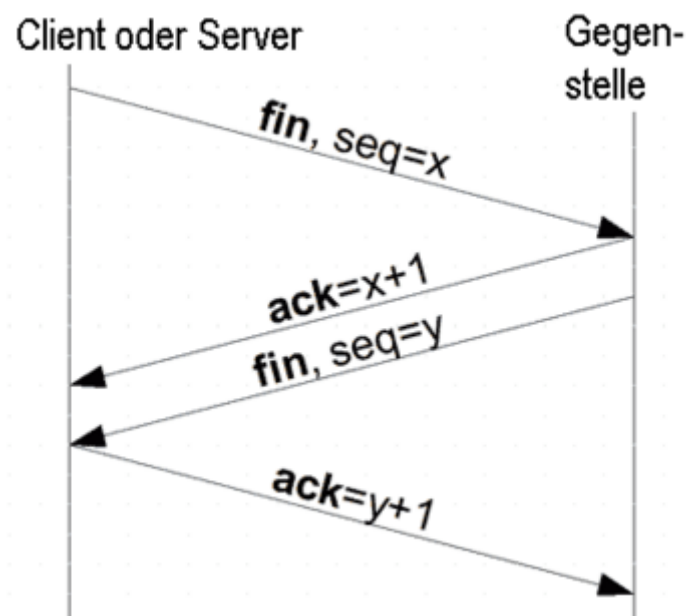
Der Sender beginnt mit dem Senden des ersten Datenpakets (Send Paket 1). Der Empfänger nimmt das Paket entgegen (Receive Paket 1) und bestätigt den Empfang (Send ACK Paket 1). Der Sender nimmt die Bestätigung entgegen (Receive ACK Paket 1) und sendet das zweite Datenpaket (Send Paket 2). Der Empfänger nimmt das zweite Paket entgegen (Receive Paket 2) und bestätigt den Empfang (Send ACK Paket 2). Der Sender nimmt die zweite Bestätigung entgegen (Receive ACK Paket 2). Und so läuft der Datenaustausch weiter, bis alle Pakete übertragen wurden.



Aufzeichnung mit Wireshark

Verbindungsabbau

Der Verbindungsabbau kann sowohl vom Client als auch vom Server vorgenommen werden. Zuerst schickt einer der beiden der Gegenstelle einen Verbindungsabbauwunsch (**FIN**). Die Gegenstelle bestätigt den Erhalt der Nachricht (ACK) und schickt gleich darauf ebenfalls einen Verbindungsabbauwunsch (**FIN**). Danach bekommt die Gegenstelle noch mitgeteilt, dass die Verbindung abgebaut ist (**ACK**).



Aufzeichnung mit Wireshark

353	4.740497	194.232.104.142	192.168.1.29	TCP	60 80 → 61868	[FIN, ACK]	Seq=130	Ack=629	Win=30464	Len=0
354	4.740526	192.168.1.29	194.232.104.142	TCP	54 61868 → 80	[ACK]	Seq=629	Ack=131	Win=65792	Len=0
355	4.740643	192.168.1.29	194.232.104.142	TCP	54 61868 → 80	[FIN, ACK]	Seq=629	Ack=131	Win=65792	Len=0
364	4.751437	194.232.104.142	192.168.1.29	TCP	60 80 → 61868	[ACK]	Seq=131	Ack=630	Win=30464	Len=0

>

Frame 355: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0

>

Ethernet II, Src: HewlettP_0b:2f:71 (24:be:05:0b:2f:71), Dst: Netgear_3e:59:a9 (b0:7f:b9:3e:59:a9)

>

Internet Protocol Version 4, Src: 192.168.1.29, Dst: 194.232.104.142

▼

Transmission Control Protocol, Src Port: 61868, Dst Port: 80, Seq: 629, Ack: 131, Len: 0

Source Port: 61868

Destination Port: 80

[Stream index: 26]

[TCP Segment Len: 0]

Sequence number: 629 (relative sequence number)

Acknowledgment number: 131 (relative ack number)

0101 = Header Length: 20 bytes (5)

> Flags: 0x011 (FIN, ACK)

Window size value: 257

[Calculated window size: 65792]

[Window size scaling factor: 256]

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4:4_09:4_09_05

Last update: 2020/03/17 08:05

UDP - User Datagram Protocol (Layer 4)

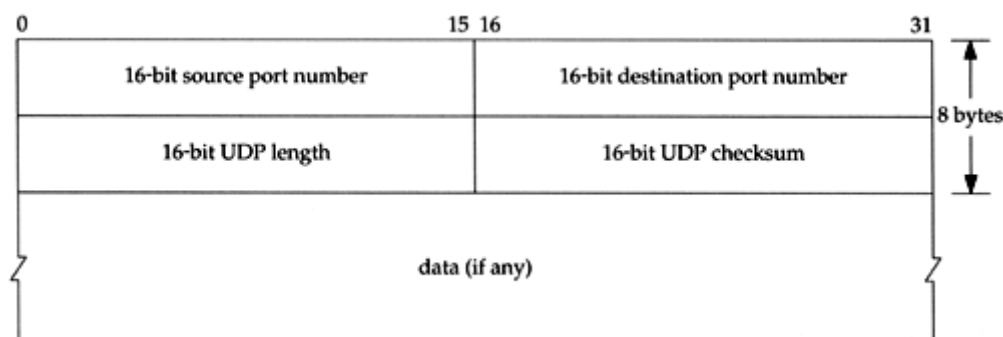
UDP ist ein **verbindungsloses Transport-Protokoll** und arbeitet auf der Schicht 4, der **Transportschicht**, des OSI-Schichtenmodells. Es hat damit eine vergleichbare Aufgabe, wie das verbindungsorientierte TCP. Allerdings arbeitet es **verbindungslos und damit unsicher**. Das bedeutet, der **Absender weiß nicht**, ob seine verschickten **Datenpakete angekommen sind**. Während TCP Bestätigungen beim Datenempfang sendet, verzichtet UDP darauf. Das hat den Vorteil, dass der Paket-Header viel kleiner ist und die Übertragungsstrecke keine Bestätigungen übertragen muss. Typischerweise wird UDP bei DNS-Anfragen, VPN-Verbindungen, Audio- und Video-Streaming verwendet.

Funktionsweise

UDP hat die selbe Aufgabe wie TCP, nur dass nahezu alle Kontrollfunktionen fehlen, dadurch schlanker und einfacher zu verarbeiten ist. So besitzt UDP keinerlei Methoden, die sicherstellen, dass ein Datenpaket beim Empfänger ankommt. Ebenso entfällt die Nummerierung der Datenpakete. UDP ist nicht in der Lage, die Datenpakete in der richtigen Reihenfolge zusammenzusetzen. Statt dessen werden die UDP-Pakete direkt an die Anwendung weitergeleitet. Für eine sichere Datenübertragung ist deshalb die Anwendung zuständig. In der Regel wird UDP für Anwendungen und Dienste verwendet, die mit Paketverlusten umgehen können oder sich selber um das Verbindungsmanagement kümmern. UDP eignet sich auch für Anwendungen, die nur einzelne, nicht zusammenhängende Datenpakete transportieren müssen.

UDP Header

UDP-Pakete setzen sich aus dem Header-Bereich und dem Daten-Bereich zusammen. Im Header sind alle Informationen enthalten, die eine einigermaßen geordnete Datenübertragung zulässt und die ein UDP-Paket als ein solches erkennen lassen. Der UDP-Header ist in 32-Bit-Blöcke unterteilt. Er besteht aus zwei solcher Blöcke, die den Quell- und Ziel-Port, die Länge des gesamten UDP-Pakets und die Check-Summe enthalten. Der UDP-Header ist mit insgesamt 8 Byte sehr schlank und lässt sich mit wenig Rechenleistung verarbeiten.





TCP vs. UDP

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4:4_09:4_09_06



Last update: **2020/03/17 07:51**

DHCP - Dynamic Host Configuration Protocol (UDP, Ports 67 + 68, Layer 7)

DHCP ist ein Protokoll, um **IP-Adressen** in einem TCP/IP-Netzwerk **zu verwalten** und an die anfragenden Hosts zu verteilen. Mit DHCP ist jeder **Netzwerk-Teilnehmer** in der Lage sich selber **automatisch zu konfigurieren**.

Warum DHCP?

Um ein Netzwerk per TCP/IP aufzubauen ist es notwendig an jedem Host eine IP-Konfiguration vorzunehmen. Für ein TCP/IP-Netzwerk müssen folgende Einstellungen an jedem Host vorgenommen werden:

- Vergabe einer eindeutigen IP-Adresse
- Zuweisen einer Subnetzmaske (Subnetmask)
- Zuweisen des zuständigen Default- bzw. Standard-Gateways
- Zuweisen des zuständigen DNS-Servers

In den ersten IP-Netzen wurden IP-Adressen noch von Hand **aufwendig vergeben** und **fest** in die Systeme **eingetragen**. Die dazu **erforderliche Dokumentation** war jedoch nicht immer fehlerfrei und schon gar nicht aktuell und vollständig. Der Ruf nach einer einfachen und automatischen Adressverwaltung wurde deshalb besonders bei Betreibern großer Netze laut. Hier war durch die manuelle Verwaltung und Konfiguration sehr **viel Planungs- und Arbeitszeit** notwendig. Um für die Betreiber der immer größer werdenden Netze eine Erleichterung zu verschaffen wurde DHCP entwickelt. Mit DHCP kann jede IP-Host die IP-Adresskonfiguration von einem DHCP-Server anfordern und sich selber automatisch konfigurieren. So müssen IP-Adressen nicht mehr manuell verwaltet und zugewiesen werden.

Funktionsweise

1. Der DHCP-Client **schickt an alle Rechner** im Netzwerk (=Broadcast) eine **DHCP-Server-Suchanfrage (DHCP-Discover)**. Bis auf den DHCP-Server verwerfen alle Rechner das Datenpaket.
2. Nur der **DHCP-Server** empfängt den DHCP-Discover und bietet dem Client mittels Broadcast eine **freie IP-Adresse an (DHCP-Offer)**
3. Nur der **DHCP-Client** empfängt den DHCP-Offer und antwortet mit einer **DHCP-Anfrage (DHCP-Request)**. Alle anderen Clients verwerfen wiederum das DHCP-Angebot.
4. Der **DHCP-Server** empfängt den DHCP-Request und **bestätigt** dem DHCP-Client die angefragte IP-Adresse **mittels DHCP-ACK**.

HugeDomains.com
Shop for Over 200,000 Premium Domains



DHCP Funktionsweise

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4:4_09:4_09_07



Last update: **2020/03/17 07:41**

DNS - Domain Name System (UDP, Port 53 , Layer 7)

Das Domain Name System, kurz DNS, wird auch als **Telefonbuch des Internets** bezeichnet. Ähnlich wie man in einem Telefonverzeichnis nach einem Namen sucht, um die Telefonnummer heraus zu bekommen, schaut man im DNS nach einem Computernamen, um die dazugehörige IP-Adresse zu bekommen. Die IP-Adresse wird benötigt, um eine Verbindung zu einem Server aufbauen zu können, über den nur der Computernamen bekannt ist.

Das Domain Name System ist ein System zur Auflösung von Computernamen in IP-Adressen und umgekehrt. DNS kennt keine zentrale Datenbank. Die Informationen sind auf vielen tausend Nameservern (DNS-Server) verteilt. Möchte man zum Beispiel die Webseite www.orf.at besuchen, dann fragt der Browser einen DNS-Server, der in der IP-Konfiguration hinterlegt ist. Das ist in der Regel der Router des Internet-Zugangs. Je nach dem, ob die DNS-Anfrage beantwortet werden kann oder nicht, wird eine Kette weiterer DNS-Server befragt, bis die Anfrage positiv beantwortet und eine IP-Adresse an den Browser zurück geliefert werden kann.

Wenn ein Computernamen oder Domain-Name nicht aufgelöst werden kann, dann kann auch keine Verbindung zu dem betreffenden Host aufgebaut werden. Es sei denn, der Nutzer verfügt über das Wissen der IP-Adresse. Das bedeutet, ohne DNS ist die Kommunikation im Netzwerk und im Internet praktisch nicht möglich. Deshalb existieren viele tausend DNS-Server auf der ganzen Welt, die zusätzlich hierarchisch angeordnet sind und sich gegenseitig über Änderungen informieren.

Top-Level-Domain & Second-Level-Domain

Domain-Namen sind hierarchisch von rechts nach links gegliedert. Der ganz rechte Abschnitt nach dem letzten Punkt heißt Top-Level-Domain (TLD), der davor Second-Level-Domain (SLD) oder einfach „Domain“. Alle weiteren Namensteile links davon sind jeweils Sub- bzw. Third-Level-Domains (Fourth Level, Fifth Level, Sixth Level usw.). Ein Beispiel verdeutlicht die Begrifflichkeiten: Der Name „www.example.com“ besteht aus drei Ebenen:

- „.com“: die erste Ebene, auch Top-Level-Domain oder Domain-Endung genannt
- „example“: die zweite Ebene, auch als Second-Level-Domain oder Domain bezeichnet
- „www“: die dritte Ebene, auch Sub- oder Third-Level-Domain genannt

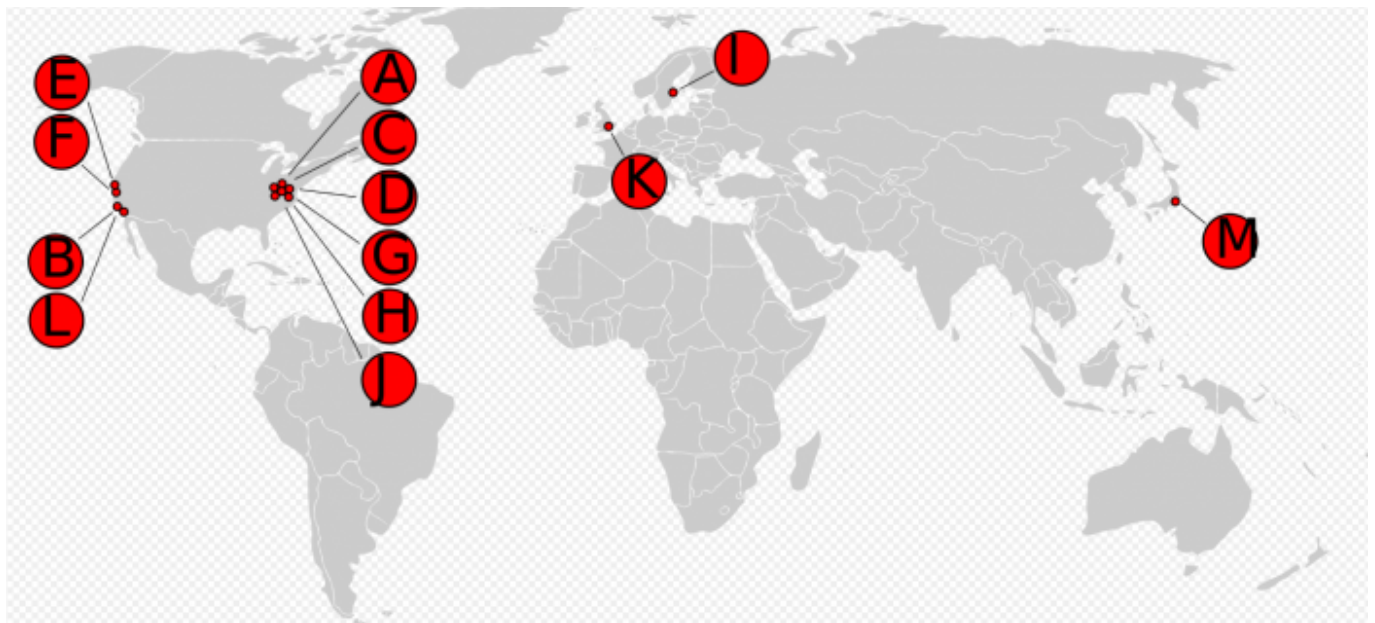
ROOT DNS Server

Root-Nameserver, kurz Root-Server, sind Server zur **Namensauflösung an der Wurzel (Root)** des Domain Name Systems im Internet. Die Zone der Root-Server umfasst Namen und **IP-Adressen aller Nameserver aller Top-Level-Domains (TLD)**.

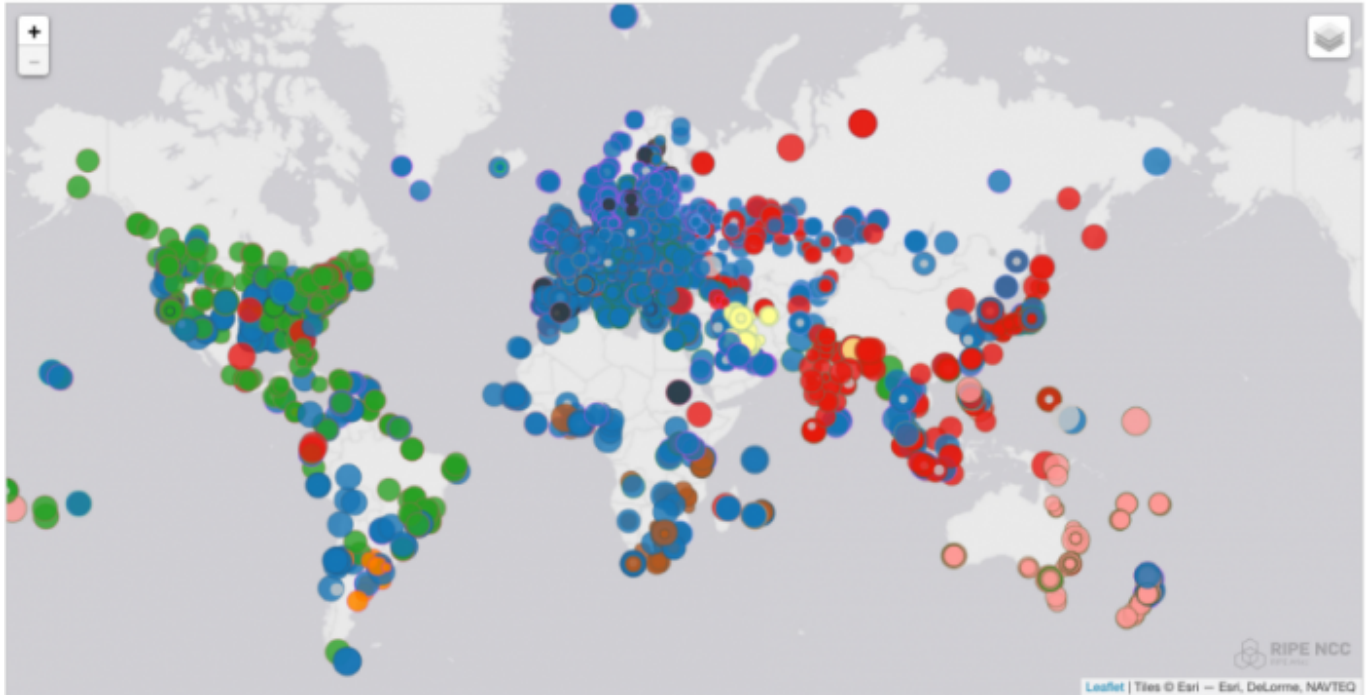
Es gibt 13 Root-Nameserver (A bis M) , die fortlaufend nach dem Schema <Buchstabe>.root-servers.net benannt sind. Jeder Root-Nameserver ist unter einer IPv4-Adresse und einer IPv6-Adresse erreichbar.

List of Root Servers

HOSTNAME	IP ADDRESSES	MANAGER
a.root-servers.net	198.41.0.4, 2001:503:ba3e::2:30	VeriSign, Inc.
b.root-servers.net	199.9.14.201, 2001:500:200::b	University of Southern California (ISI)
c.root-servers.net	192.33.4.12, 2001:500:2::c	Cogent Communications
d.root-servers.net	199.7.91.13, 2001:500:2d::d	University of Maryland
e.root-servers.net	192.203.230.10, 2001:500:a8::e	NASA (Ames Research Center)
f.root-servers.net	192.5.5.241, 2001:500:2f::f	Internet Systems Consortium, Inc.
g.root-servers.net	192.112.36.4, 2001:500:12::d0d	US Department of Defense (NIC)
h.root-servers.net	198.97.190.53, 2001:500:1::53	US Army (Research Lab)
i.root-servers.net	192.36.148.17, 2001:7fe::53	Netnod
j.root-servers.net	192.58.128.30, 2001:503:c27::2:30	VeriSign, Inc.
k.root-servers.net	193.0.14.129, 2001:7fd::1	RIPE NCC
l.root-servers.net	199.7.83.42, 2001:500:9f::42	ICANN
m.root-servers.net	202.12.27.33, 2001:dc3::35	WIDE Project



Alle Root-Nameserver setzen Anycast zur Lastverteilung ein, sodass die 13 Adressen von tatsächlich mehreren hundert Servern an verschiedenen Orten der Welt bedient werden.



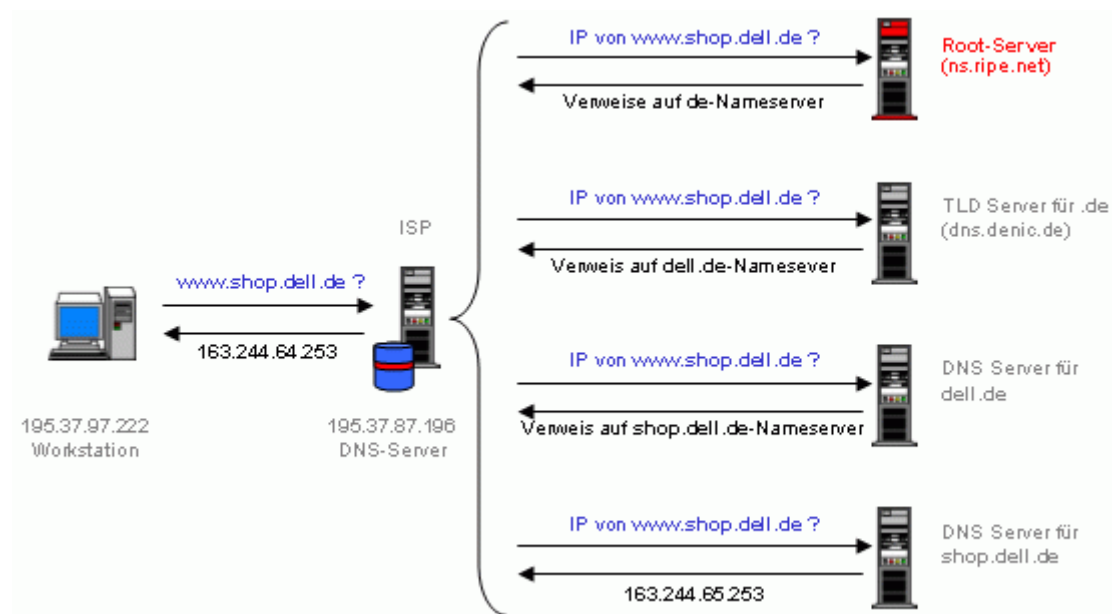
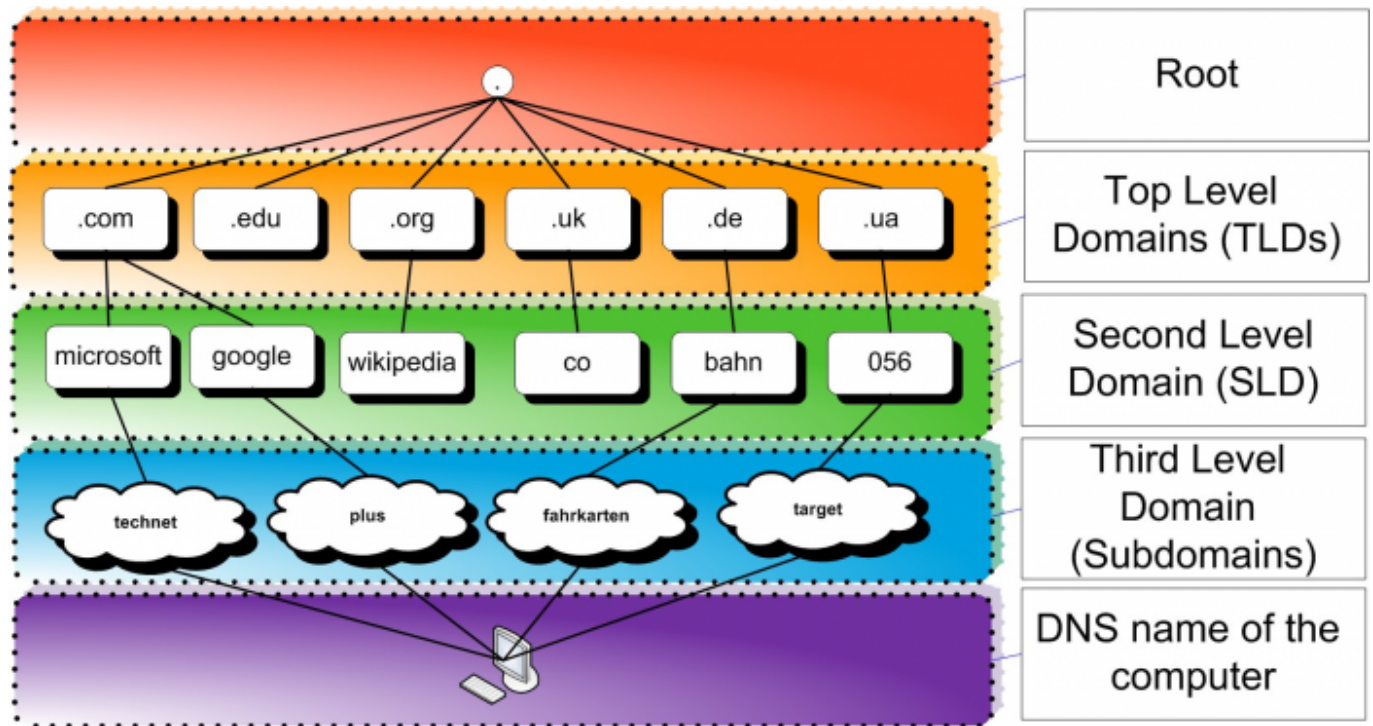
FQDN (Fully Qualified Domain Name)

Der vollständige Name einer Domain wird als ihr Fully Qualified Domain Name (FQDN) bezeichnet. Der Domain-Name ist in diesem Fall eine absolute Adresse.

Der FQDN www.example.com ergibt sich durch:

```
3rd-level-label. 2nd-level-label. Top-Level-Domain. root-label
```

Da das Root-Label immer leer ist (es besteht aus einer leeren Zeichenkette), wird bei den meisten Benutzer-Anwendungen (zum Beispiel Browsern) in der Regel auf die Eingabe des Punktes zwischen dem Label der Top Level Domain und dem root-label verzichtet. Streng genommen handelt es sich bei dieser Schreibweise nicht mehr um eine absolute, sondern um eine relative Adresse und damit nicht mehr um einen FQDN.





Video

Beispiel zur rekursiven Namensauflösung in Filius

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

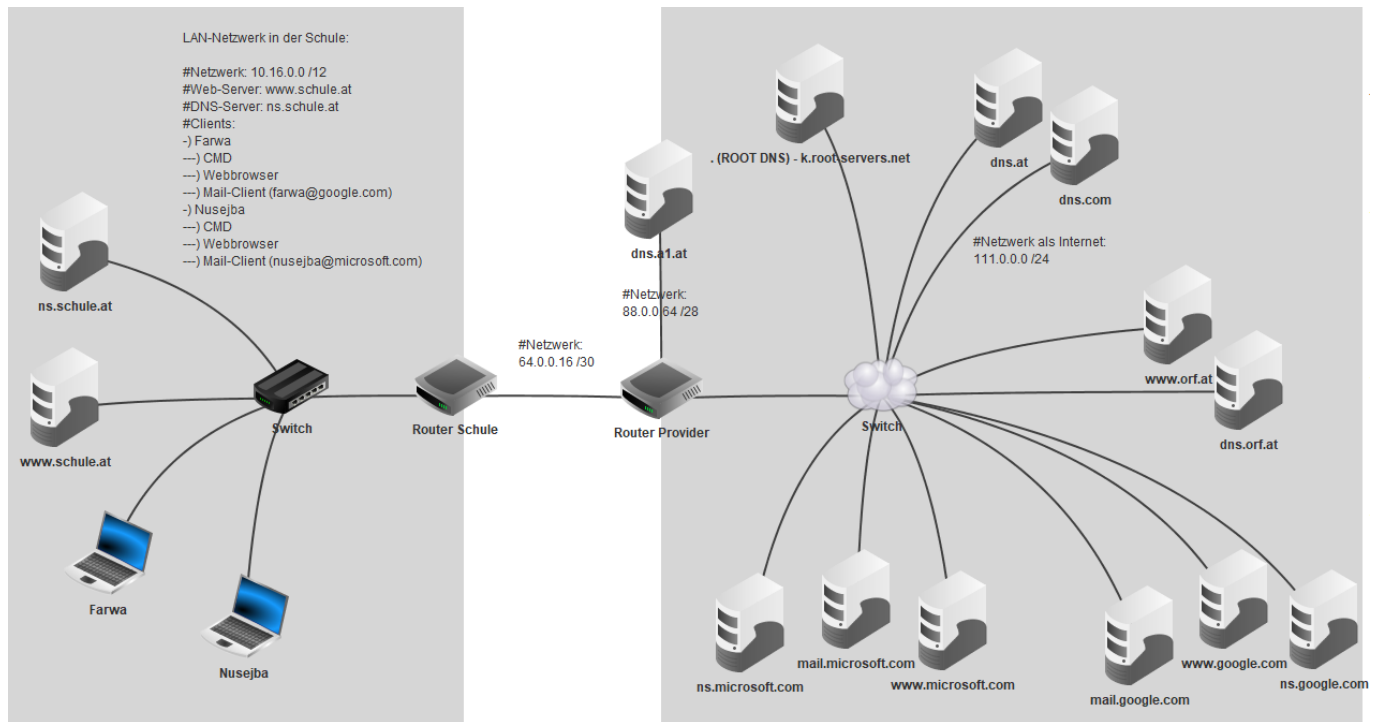
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4:4_09:4_09_08



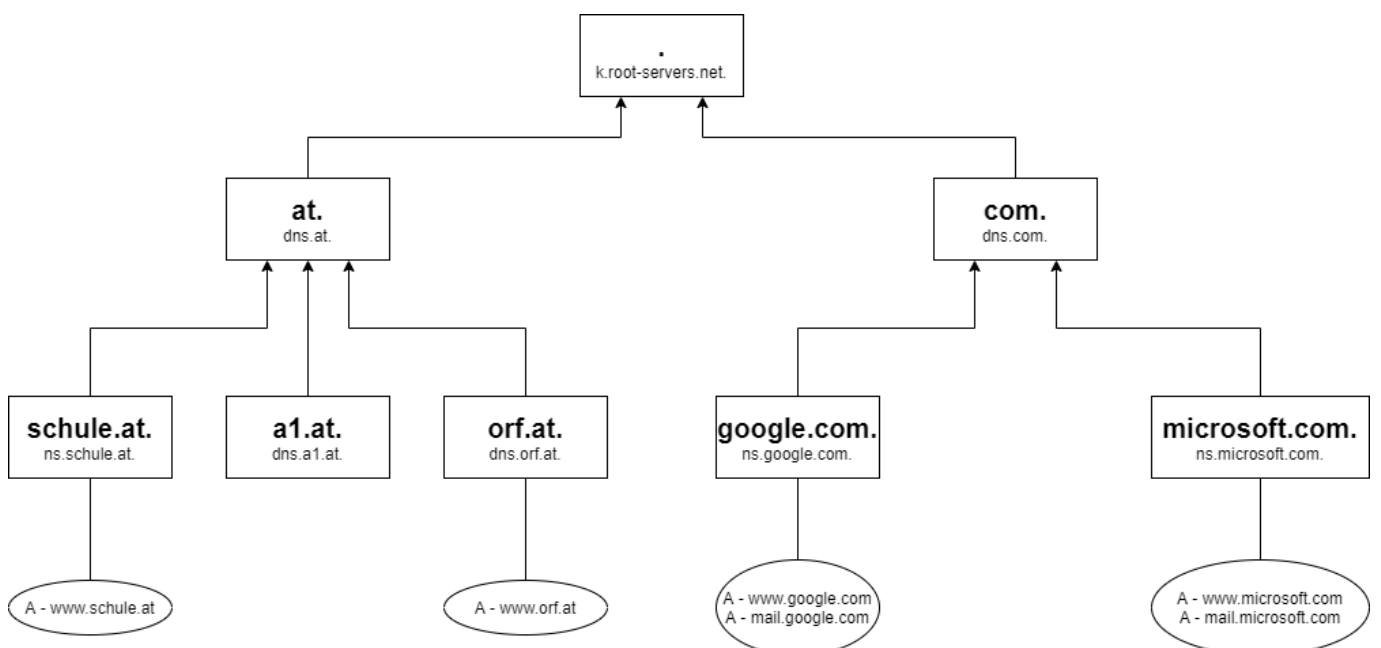
Last update: **2020/03/27 11:12**

FILIUS-BEISPIEL zur rekursiven Namensauflösung

NW-Übersicht



DNS-Hierarchie



. (ROOT) - k.root-servers.net

. (ROOT DNS) - k.root-servers.net - 111.0.0.1

DNS-Server

☒ Aktiviere rekursive Domain-Auflösung

Domainname:

IP-Adresse:

Domainname	IP-Adresse
dns.at.	111.0.0.10
dns.com.	111.0.0.20
k.root-servers.net.	111.0.0.1

. (ROOT DNS) - k.root-servers.net - 111.0.0.1

DNS-Server

☒ Aktiviere rekursive Domain-Auflösung

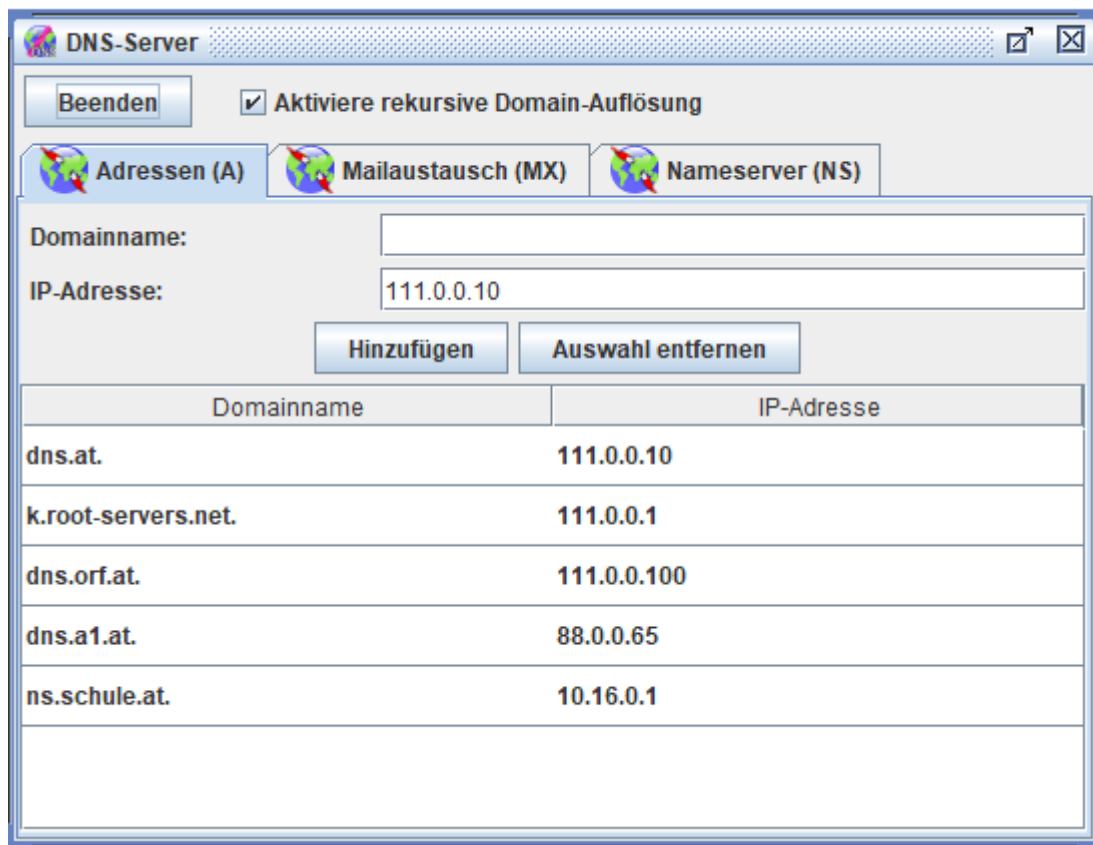
Domain:

Nameserver:

Domain	Nameserver
at.	dns.at.
com.	dns.com.
.	k.root-servers.net.

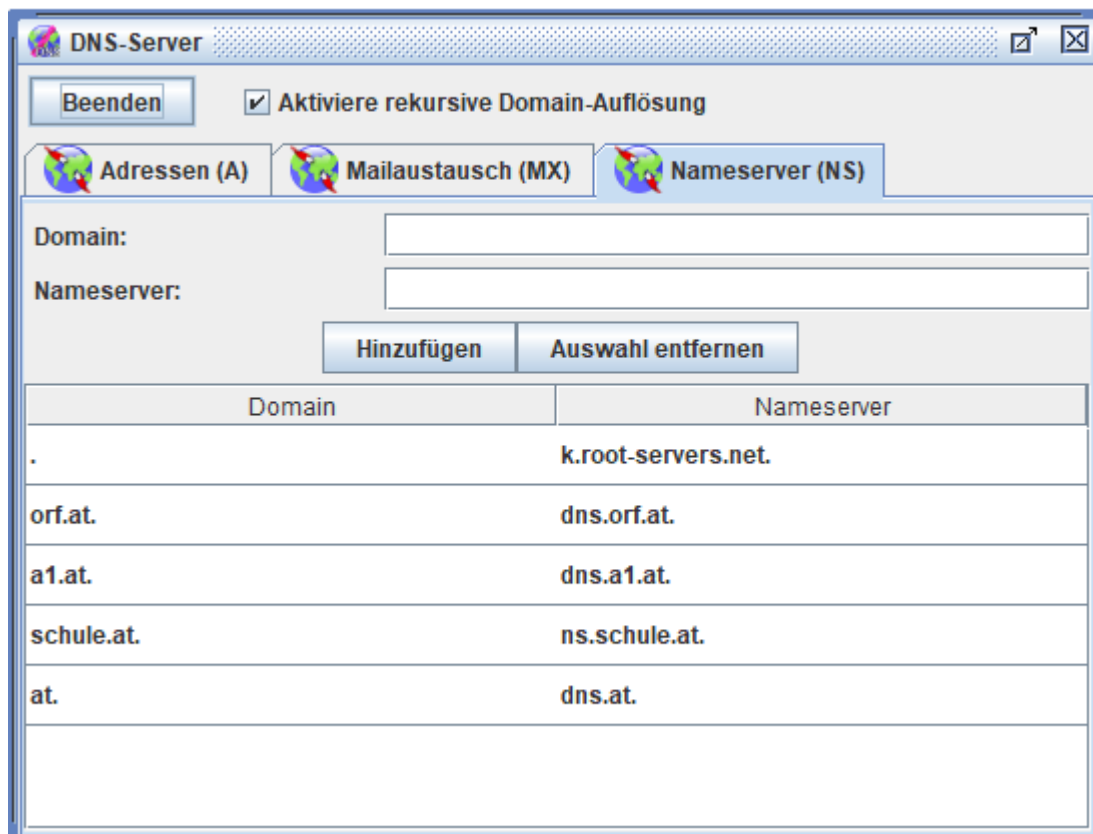
dns.at

dns.at - 111.0.0.10



Domainname	IP-Adresse
dns.at.	111.0.0.10
k.root-servers.net.	111.0.0.1
dns.orf.at.	111.0.0.100
dns.a1.at.	88.0.0.65
ns.schule.at.	10.16.0.1

dns.at - 111.0.0.10



Domain	Nameserver
.	k.root-servers.net.
orf.at.	dns.orf.at.
a1.at.	dns.a1.at.
schule.at.	ns.schule.at.
at.	dns.at.

dns.com

dns.com - 111.0.0.20

DNS-Server

☒ Aktiviere rekursive Domain-Auflösung

☒ Adressen (A) ☐ Mailaustausch (MX) ☐ Nameserver (NS)

Domainname:

IP-Adresse:

Domainname	IP-Adresse
dns.com.	111.0.0.20
k.root-servers.net.	111.0.0.1
ns.google.com.	111.0.0.200
ns.microsoft.com.	111.0.0.250

dns.com - 111.0.0.20

DNS-Server

☒ Aktiviere rekursive Domain-Auflösung

☐ Adressen (A) ☐ Mailaustausch (MX) ☒ Nameserver (NS)

Domain:

Nameserver:

Domain	Nameserver
.	k.root-servers.net.
google.com.	ns.google.com.
com.	dns.com.
microsoft.com.	ns.microsoft.com.

ns.schule.at

ns.schule.at - 10.16.0.1

DNS-Server

☒ Aktiviere rekursive Domain-Auflösung

☒ Adressen (A) ☐ Mailaustausch (MX) ☐ Nameserver (NS)

Domainname:

IP-Adresse:

Domainname	IP-Adresse
ns.schule.at.	10.16.0.1
www.schule.at.	10.16.0.2
k.root-servers.net.	111.0.0.1

ns.schule.at - 10.16.0.1

DNS-Server

☒ Aktiviere rekursive Domain-Auflösung

☐ Adressen (A) ☐ Mailaustausch (MX) ☒ Nameserver (NS)

Domain:

Nameserver:

Domain	Nameserver
.	k.root-servers.net.
schule.at.	ns.schule.at.

dns.a1.at

dns.a1.at - 88.0.0.65

DNS-Server

☐ Aktiviere rekursive Domain-Auflösung

Domainname:

IP-Adresse:

Domainname	IP-Adresse
dns.a1.at.	88.0.0.65
k.root-servers.net.	111.0.0.1

dns.a1.at - 88.0.0.65

DNS-Server

☐ Aktiviere rekursive Domain-Auflösung

Domain:

Nameserver:

Domain	Nameserver
.	k.root-servers.net.
a1.at.	dns.a1.at.

dns.orf.at

dns.orf.at - 111.0.0.100

DNS-Server

☒ Aktiviere rekursive Domain-Auflösung

☒ Adressen (A) ☐ Mailaustausch (MX) ☐ Nameserver (NS)

Domainname:

IP-Adresse:

Domainname	IP-Adresse
dns.orf.at.	111.0.0.100
www.orf.at.	111.0.0.101
k.root-servers.net.	111.0.0.1

dns.orf.at - 111.0.0.100

DNS-Server

☒ Aktiviere rekursive Domain-Auflösung

☒ Adressen (A) ☐ Mailaustausch (MX) ☐ Nameserver (NS)

Domain:

Nameserver:

Domain	Nameserver
.	k.root-servers.net.
orf.at.	dns.orf.at.

ns.google.com

ns.google.com - 111.0.0.200

DNS-Server

☒ Aktiviere rekursive Domain-Auflösung

☒ Adressen (A) ☐ Mailaustausch (MX) ☐ Nameserver (NS)

Domainname:

IP-Adresse:

Domainname	IP-Adresse
ns.google.com.	111.0.0.200
www.google.com.	111.0.0.201
k.root-servers.net.	111.0.0.1
mail.google.com.	111.0.0.202

ns.google.com - 111.0.0.200

DNS-Server

☒ Aktiviere rekursive Domain-Auflösung

☒ Adressen (A) ☐ Mailaustausch (MX) ☒ Nameserver (NS)

Domain:

Nameserver:

Domain	Nameserver
google.com.	ns.google.com.
.	k.root-servers.net.

ns.microsoft.com

ns.microsoft.com - 111.0.0.250

DNS-Server

☒ Aktiviere rekursive Domain-Auflösung

☒ Adressen (A) ☐ Mailaustausch (MX) ☐ Nameserver (NS)

Domainname:

IP-Adresse:

Domainname	IP-Adresse
ns.microsoft.com.	111.0.0.250
www.microsoft.com.	111.0.0.251
mail.microsoft.com.	111.0.0.252
k.root-servers.net.	111.0.0.1

ns.microsoft.com - 111.0.0.250

DNS-Server

☒ Aktiviere rekursive Domain-Auflösung

☒ Adressen (A) ☐ Mailaustausch (MX) ☒ Nameserver (NS)

Domain:

Nameserver:

Domain	Nameserver
microsoft.com.	ns.microsoft.com.
.	k.root-servers.net.

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4:4_09:4_09_08:4_09_08_01

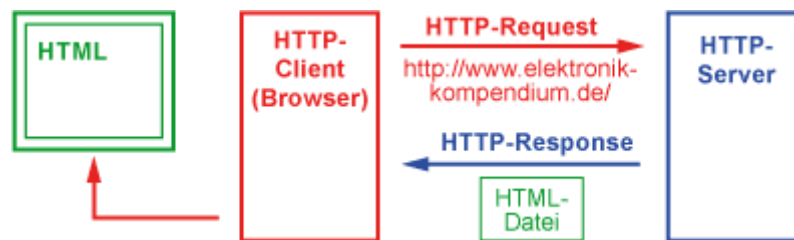


Last update: **2020/03/27 11:33**

Hypertext Transfer Protocol (HTTP) (Port 80 - Layer 7)

HTTP ist das Kommunikationsprotokoll im World Wide Web (WWW). Die wichtigsten Funktionen sind Dateien vom Webserver anzufordern und in den Browser zu laden. Der Browser übernimmt dann die Darstellung von Texten und Bildern und kümmert sich um das Abspielen von Audio- und Video-Daten.

Funktionsweise



Die Kommunikation findet nach dem **Client-Server-Prinzip** statt. Der **HTTP-Client (Browser)** sendet seine **Anfrage (HTTP-Request)** an den **HTTP-Server (Webserver/Web-Server)**. Dieser bearbeitet die Anfrage und schickt seine **Antwort (HTTP-Response)** zurück. Nach der Antwort durch den Server ist diese Verbindung beendet. Typischerweise finden gleichzeitig mehrere HTTP-Verbindungen statt.

HTTP Adressierung

Damit der Server weiß, was er dem HTTP-Client schicken soll, adressiert der HTTP-Client eine Datei, die sich auf dem HTTP-Server befinden muss. Dazu wird vom HTTP-Client ein **URL (Uniform Resource Locator)** im HTTP-Header an den HTTP-Server übermittelt:

```
http://Servername.Domainname.Top-Level-Domain:TCP-Port/Pfad/Datei
```

z. B.

```
http://www.elektronik-kompendium.de:80/sites/kom/0902231.html
```

HTTP Request

Der HTTP-Request ist die Anfrage des HTTP-Clients an den HTTP-Server. Ein HTTP-Request besteht aus den Angaben **Methode, URL und dem Request-Header**. Die häufigsten Methoden sind **GET und POST**. Dahinter folgt durch ein Leerzeichen getrennt der URL und die verwendete HTTP-Version. In weiteren Zeilen folgt der Header und bei der Methode POST durch eine Leerzeile (!) getrennt die Formular-Daten.

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4:4_09:4_09_09



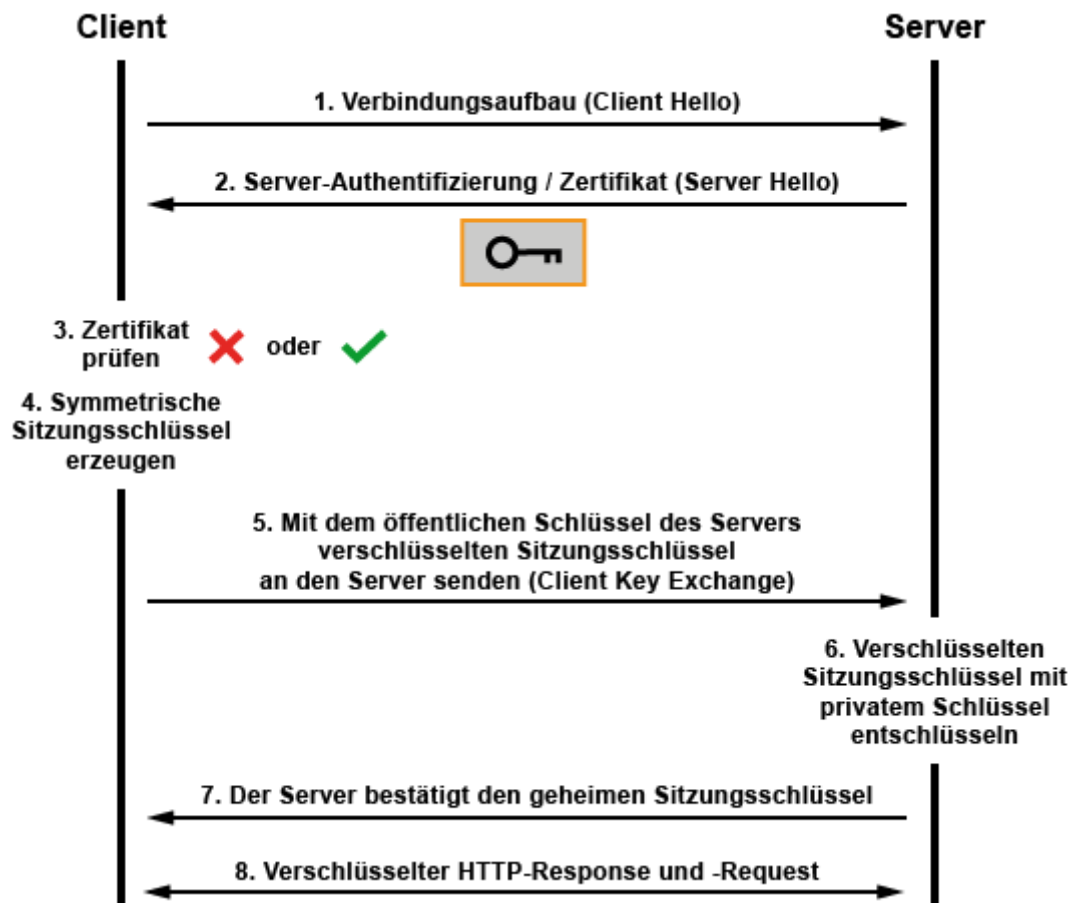
Last update: **2020/03/17 07:41**

HyperText Transfer Protocol Secure (HTTPS) (Port 443 - Layer 7)

HTTPS bzw. HTTP Secure ist die Anwendung von HTTP in **Verbindung mit Verschlüsselung und Authentifizierung**. Wobei in der Regel nur der **angefragte Webserver sich mit einem Zertifikat authentisieren** muss.

Eine **verschlüsselte Verbindung mit einem Browser** signalisiert man mit einem „https:“ (TCP-Port 443) statt „http:“ (TCP-Port 80). Dabei muss sich der Webserver dem Client gegenüber authentisieren, ob er tatsächlich der Webserver ist, der sich unter der eingegebenen Adresse befindet. Zusätzlich wird die **Verbindung bzw. Sitzung Ende-zu-Ende-verschlüsselt**. Das bedeutet, die **Stationen zwischen Client und Server** können die Kommunikation **nicht entschlüsseln**.

Für die Authentifizierung und Verschlüsselung ist SSL/TLS verantwortlich. Es schiebt sich zwischen HTTP und dem Transportprotokoll TCP. Damit steht SSL/TLS auch für andere Anwendungsprotokolle zur Verfügung. Beispielsweise SMTPS, IMAPS und FTPS. SSL arbeitet für den Anwender nahezu unsichtbar.



1. Client Hello: Der Client kontaktiert den Server über ein Protokoll mit Verschlüsselungsoptionen.
2. Server Hello, Certificate, Server Key Exchange, Server Hello Done: Der Server nimmt die Verbindung an und schickt sein Zertifikat mit dem öffentlichen Schlüssel seines Schlüsselpaares zur Authentifizierung an den Client.
3. Der Client überprüft das Server-Zertifikat und dessen Gültigkeit (Validierung). Erkennt der Client das Zertifikat als ungültig wird die Verbindung an dieser Stelle abgebrochen.
4. Erkennt der Client das Zertifikat als gültig erzeugt der Client den symmetrischen Sitzungsschlüssel.

5. Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message: Mit dem öffentlichen Schlüssel des Servers verschlüsselt der Client den Sitzungsschlüssel und schickt ihn an den Server.
6. Mit seinem privaten Schlüssel kann der Server den verschlüsselten Sitzungsschlüssel entschlüsseln.
7. Encrypted Handshake Message, Change Cipher Spec, Encrypted Handshake Message: Der Server bestätigt den geheimen Sitzungsschlüssel.
8. Danach werden alle HTTP-Requests und -Responses verschlüsselt, bis die Verbindung abgebaut wird.

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4:4_09:4_09_10

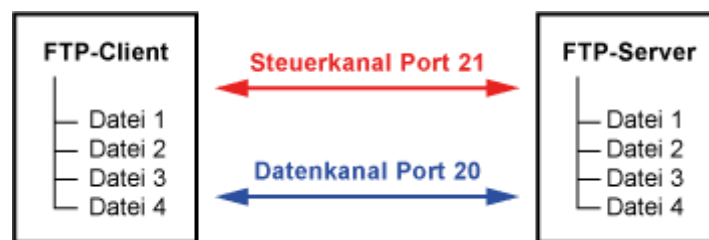


Last update: **2020/03/17 07:42**

File Transfer Protocol (FTP) (Port 20 und 21 - Layer 7)

FTP ist ein Kommunikationsprotokoll, um Dateien zwischen unterschiedlichen Computersystemen zu übertragen. Die Übertragung findet nach dem Client-Server-Prinzip statt. Ein FTP-Server stellt dem FTP-Client Dateien zur Verfügung. Der FTP-Client kann Dateien auf dem FTP-Server ablegen, löschen oder herunterladen. Mit einem komfortablen FTP-Client arbeitet man ähnlich, wie mit einem Dateimanager.

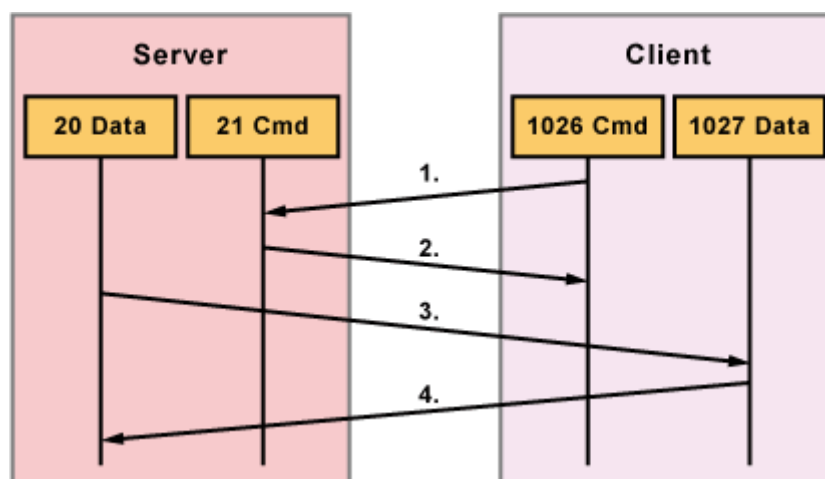
Funktionsweise

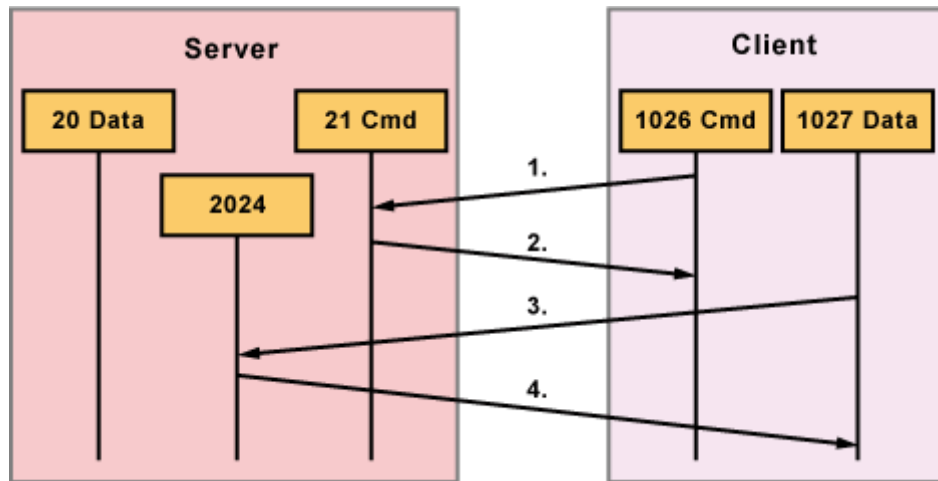


Die Kommunikation findet nach dem Client-Server-Prinzip statt. Wobei FTP zwischen Client und Server zwei logische Verbindungen herstellt. Eine Verbindung ist der Steuerkanal (command channel) über den TCP-Port 21. Dieser Kanal dient ausschließlich zur Übertragung von FTP-Kommandos und deren Antworten. Die zweite Verbindung ist der Datenkanal (data channel) über den TCP-Port 20. Dieser Kanal dient ausschließlich zur Übertragung von Daten. Über den Steuerkanal tauschen Client und Server Kommandos aus, die eine Datenübertragung über den Datenkanal einleiten und beenden.

Aktives vs. Passives FTP

Der FTP-Verbindungsaufbau sieht vor, dass der Steuerkanal vom FTP-Client zum FTP-Server aufgebaut wird. Steht der Steuerkanal wird der Datenkanal vom FTP-Server zum FTP-Client initiiert (aktives FTP). Befindet sich der FTP-Client hinter einem NAT-Router oder einer Firewall und verfügt parallel dazu nur über eine private IPv4-Adresse, dann kommt die Verbindung nicht zustande. Die Verbindungsanforderung vom Server an den Client wird von der Firewall bzw. dem Router abgeblockt, bzw. kann wegen der privaten IPv4-Adresse gar nicht geroutet werden. Für diesen Fall gibt es das passive FTP, bei dem auch der Client den Datenkanal initiiert.





Am Anfang jeder FTP-Verbindung steht die Authentifizierung des Benutzers. Danach erfolgt der Aufbau des Steuerkanals über Port 21 und des Datenkanals über Port 20. Wenn die Dateiübertragungen abgeschlossen sind, werden die Verbindungen vom Benutzer oder vom Server (Timeout) beendet.



Aktives vs. Passives FTP

From:
<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4:4_09:4_09_11

Last update: **2020/03/17 07:42**



Secure Shell (SSH) (Port 22 - Layer 7)

SSH bzw. Secure Shell ist ein **kryptografisches Protokoll** mit dem man auf einen entfernten Rechner mittels einer **verschlüsselten Verbindung über ein unsicheres Netzwerk** zugreifen kann.

Die Shell (Kommandozeile) bietet **vollen Zugriff auf das Dateisystem und alle Funktionen** des Rechners.

Die Funktionen der Secure Shell beinhalten den Login auf entfernte Rechner, die interaktive und nicht interaktive Ausführung von Kommandos und das Kopieren von Dateien zwischen verschiedenen Rechnern eines Netzwerks. SSH bietet dazu eine kryptografisch gesicherte Kommunikation über das unsichere Netzwerk, eine zuverlässige gegenseitige Authentisierung, Verschlüsselung des gesamten Datenverkehrs auf Basis eines Passworts oder Public/Private-Key-Login-Methoden

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4:4_09:4_09_12

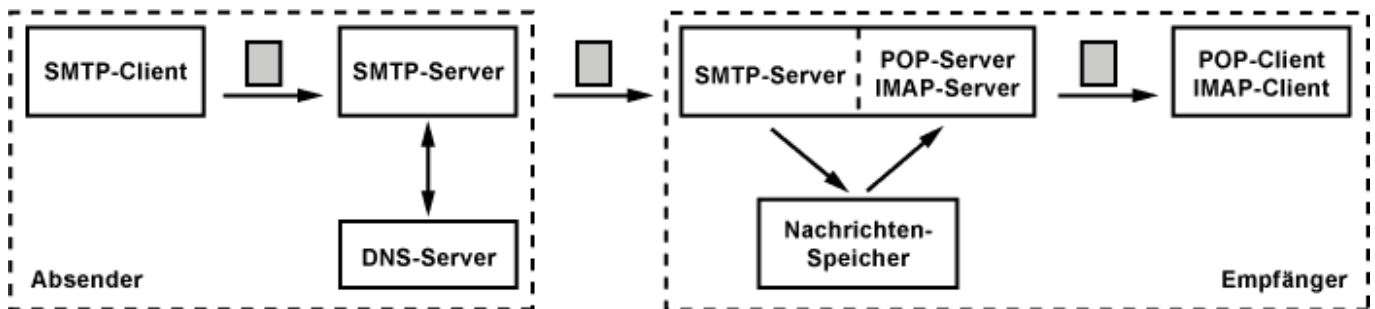


Last update: **2020/03/17 07:42**

Simple Mail Transfer Protocol (SMTP) (Port 25 - Layer 7)

SMTP ist ein **Kommunikationsprotokoll für die Übertragung von E-Mails**. Die **Kommunikation** erfolgt **zwischen** einem **E-Mail-Client und einem SMTP-Server (Postausgangsserver)** oder zwischen zwei SMTP-Server. Für den **Austausch** der E-Mails sind die **Mail Transfer Agents (MTAs)** zuständig. Untereinander verständigen sich die MTAs mit dem SMTP-Protokoll.

Neben SMTP gibt mit POP und IMAP noch zwei weitere Protokolle für den E-Mail-Austausch. Diese beiden Protokolle dienen jedoch nur dazu, um E-Mail abzuholen oder online zu verwalten. SMTP dagegen ist ein Kommunikationsprotokoll, das **E-Mails entgegennehmen und weiterleiten kann**.



Der Ablauf des E-Mail-Routings sieht in etwa so aus: Der SMTP-Server fragt einen DNS-Server ab und erhält eine Aufstellung von Mail-Servern, die E-Mails für den Ziel-SMTP-Server entgegennehmen. Jeder dieser Mail-Server (Mail Exchange) ist mit einer Priorität versehen. Der SMTP-Server versucht die Mail-Server in der vorgegebenen Reihenfolge zu kontaktieren, um die E-Mail zu übermitteln.

Nachteile

- Für versendete E-Mails keine Versandbestätigung
- Geht eine E-Mail verloren, werden weder Sender noch Empfänger darüber informiert
- Nicht vorhandene Authentisierung des Benutzers beim Verbindungsaufbau zwischen SMTP-Client und SMTP-Server. Das führt dazu, dass eine beliebige Absenderadresse beim Versand einer E-Mail angegeben werden kann

Simple Mail Transfer Protocol Secure (SMTPS) (Port 465 und 587- Layer 7)

SMTPS (Simple Mail Transfer Protocol Secure) bezeichnet ein Verfahren zur **Absicherung der Kommunikation** beim E-Mail-Transport via **SMTP über SSL/TLS** und ermöglicht dadurch **Authentifizierung der Kommunikationspartner** auf Transportebene sowie **Integrität und Vertraulichkeit** der übertragenen Nachrichten.

SMTPS ist **kein eigenes Protokoll** und auch keine Erweiterung von SMTP, da es **vollkommen transparent und unabhängig** von diesem auf der Transportschicht arbeitet.

Das bedeutet, dass die Verbindung, über die SMTP abgewickelt wird, softwaremäßig mit den Verfahren **SSL oder TLS abgesichert** wird. Dies geschieht direkt beim Verbindungsaufbau, noch bevor irgendwelche Maildaten ausgetauscht werden. Da also die Verwendung der Sicherungsschicht nicht verhandelt wird, sind SMTPS-Dienste in der Regel auf einem eigenen TCP-Port erreichbar.

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

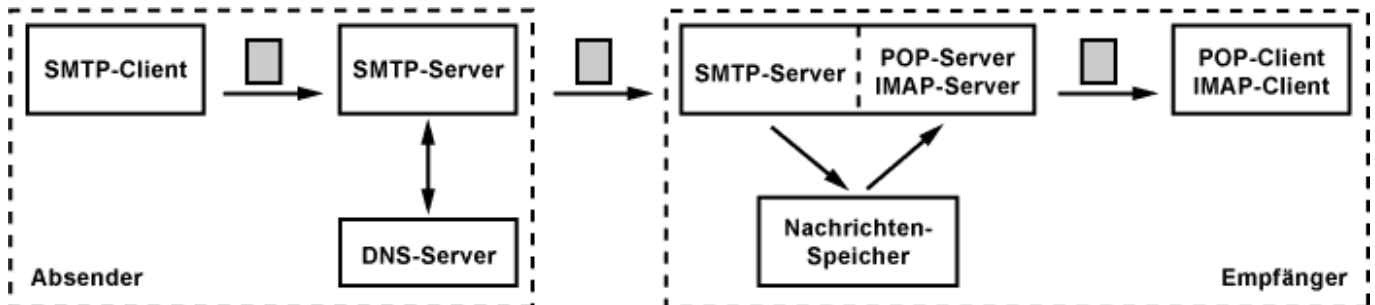
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4:4_09:4_09_13



Last update: **2020/03/17 07:45**

Post Office Protocol Version 3 (POP3) (Port 110 - Layer 7)

POP ist ein Kommunikationsprotokoll, um **E-Mails von einem Posteingangsserver (POP-Server) abzuholen**. Die **Kommunikation** erfolgt **zwischen** einem **E-Mail-Client** und einem **E-Mail-Server (Posteingangsserver)**. Das Protokoll, das diesen Zugriff regelt, nennt sich POP (aus dem Jahr 1984), das in der aktuellen Version 3 vorliegt, und deshalb manchmal auch als POP3 bezeichnet wird.



Per Fernzugriff werden die gespeicherten E-Mails abgerufen und auf dem lokalen Computer gespeichert. POP sieht das Prinzip der Offline-Verarbeitung von E-Mails vor. Online werden die E-Mails vom Posteingangsserver vom E-Mail-Client heruntergeladen. Wenn sich darunter E-Mails mit einem großen Dateianhang befinden, kann der Download schon mal etwas länger dauern. Erst nach erfolgreichem und vollständigem Zugriff werden die E-Mails auf dem Server gelöscht. Die Bearbeitung der eingegangenen E-Mails erfolgt anschließend auf dem lokalen Computer des Benutzers ohne Verbindung (offline) POP-Server.

Die Verbindung zwischen POP-Server und E-Mail-Client erfolgt über TCP auf Port 110.

Post Office Protocol Version 3 Secure (POP3S) (Port 995 - Layer 7)

POP3S bezeichnet ein Netzwerkprotokoll zur Erweiterung des E-Mail-Übertragungsprotokolls POP3 um eine Verschlüsselung durch SSL/TLS. Üblicherweise wird für POP3S TCP auf Port 995 genutzt.

From:
<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

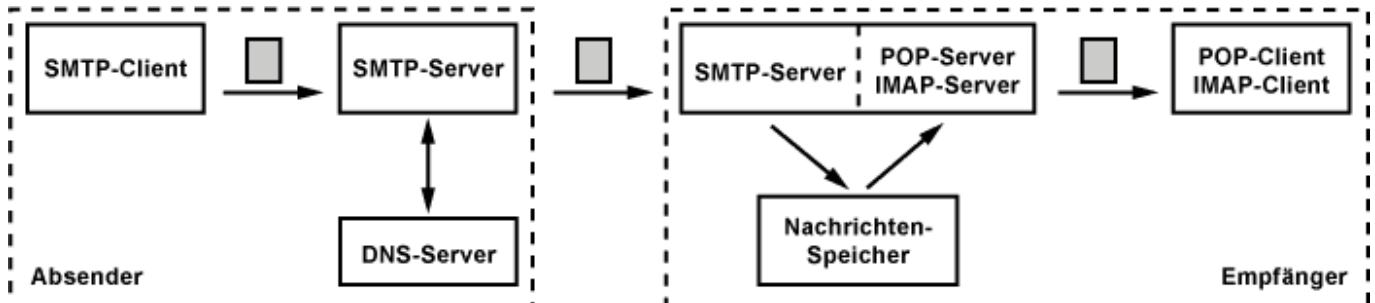
Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4:4_09:4_09_14

Last update: 2020/03/17 07:45



Internet Mail Access Protocol (IMAP) (TCP-Port 143 - Layer 7)

IMAP ist ein Kommunikationsprotokoll, um **E-Mails** auf einem **entfernten Server** ähnlich wie Dateien **zu verwalten**. Dabei **bleiben alle E-Mails auf dem IMAP-Server**. Erst wenn eine E-Mail gelesen werden soll, wird sie heruntergeladen.



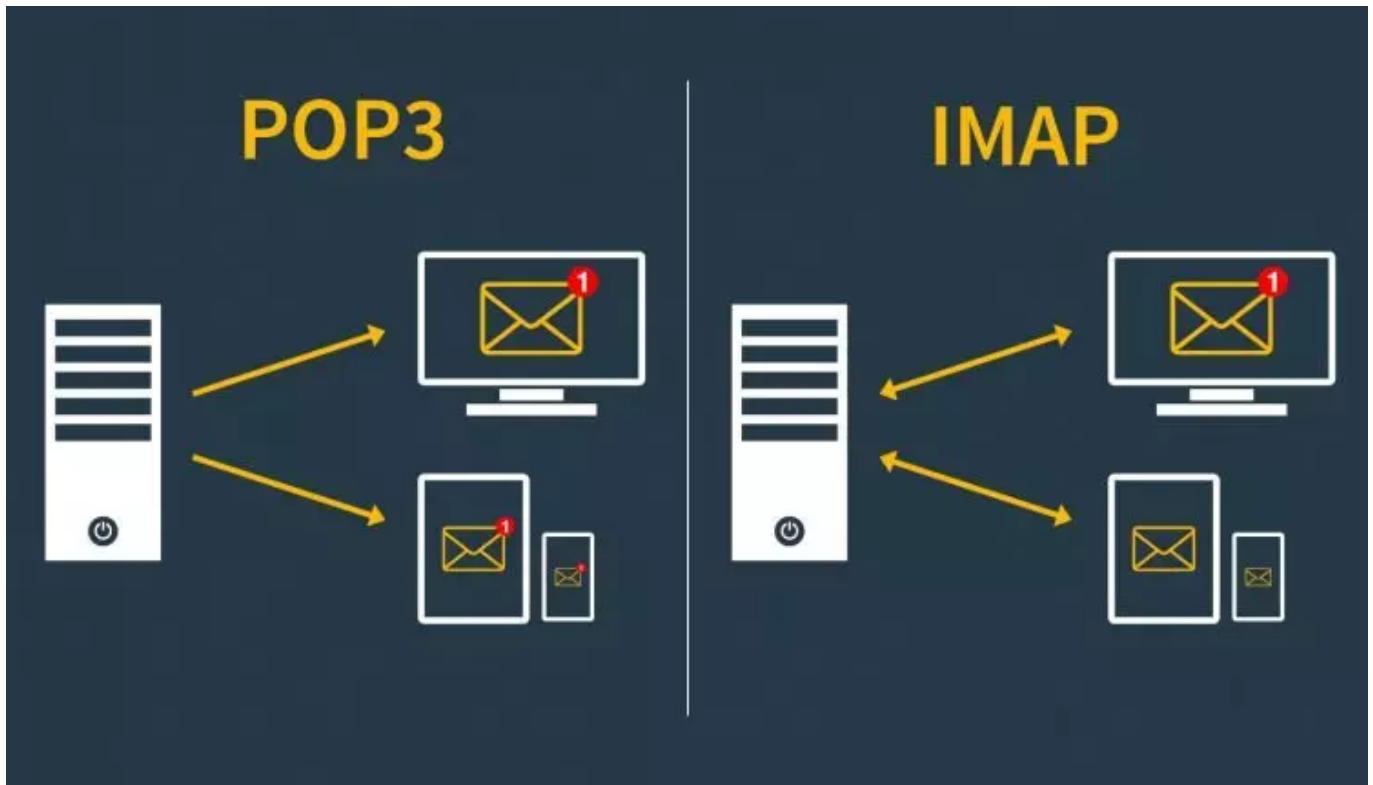
IMAP erlaubt den Zugriff auf eine Mailbox, **ähnlich wie mit POP**. Der entscheidende **Unterschied** zwischen beiden Protokollen ist der **Online-Modus von IMAP**, über den der **E-Mail-Client ständig in Verbindung mit dem E-Mail-Server** steht. Während einer IMAP-Sitzung kann auf einzelne E-Mails zugegriffen werden, die so lange auf dem Server bleiben, bis sie gelöscht werden. Dadurch kann von überall auf dem Server zugegriffen werden. Auch mit einem Endgerät, das nur mit geringer Bandbreite am Netzwerk angeschlossen ist. Die **E-Mails werden nur dann heruntergeladen, wenn der Anwender diese Lesen will**. E-Mails mit einem großen Dateianhang verstopfen dann nicht mehr ungewollt den Zugang zum Netzwerk.

Internet Mail Access Protocol Secure (IMAPS) (TCP-Port 143 - Layer 7)

Bei der Verwendung von IMAPS wird die Verbindung zum Server bereits während des Verbindungsaufbaus durch SSL verschlüsselt. Damit der Server das erkennt, muss ein anderer Port verwendet werden. Dafür wurde der Port 993 reserviert.

Nach dem Aufbau der SSL-Verbindung wird IMAP verwendet. Die SSL-Schicht ist für das IMAP-Protokoll transparent, d. h., es werden keine Änderungen am IMAP-Protokoll vorgenommen.

IMAP vs POP3



POP3 entstammt noch einer Zeit, in der Mails mit einem Rechner – ob zu Hause oder auf der Arbeit – abgerufen wurden. Viele Menschen greifen heutzutage allerdings mit zahlreichen Geräten (Arbeitsplatz-Rechner, Tablet, Rechner zu Hause, Smartphone,...) auf ihre Mails zurück. Entsprechend offenbaren sich die Schwächen der POP3-Technologie beim Versuch Posteingang und Postausgang auf verschiedenen Geräten in Einklang zu bringen.

Hat der Rechner im Büro die Mails bereits abgerufen und vom Server gelöscht, wird das Tablet mitteilen, es gebe keine neuen Mails. Ein klassischer „Workaround“ wäre zum Beispiel dem Rechner im Büro mitzuteilen, er soll die Mails zwar abrufen, aber er sollte diese bitte auf dem Server belassen. Dann wiederum werden Mails auf allen Geräten als neu betrachtet und entsprechend abgerufen.

Ebenfalls lästig: Markiert man eine Mail als gelesen, passiert dies nur auf dem jeweils einen Gerät. Alle anderen Geräte kriegen davon schlichtweg nichts mit.

Und: Greift man gerne auf seine „gesendeten Mails“ zurück, wird man ebenfalls feststellen, dass nur die Mails gespeichert werden, die auf dem entsprechenden Gerät verschickt worden sind. Das gleiche gilt im Übrigen auch für die Entwürfe. Mal eben eine Mail im Büro anfangen, in den Entwürfen speichern und auf dem Weg nach Hause in der Bahn via Smartphone beenden und verschicken...geht halt nicht.

From:
<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4:4_09:4_09_15

Last update: **2020/03/17 07:49**

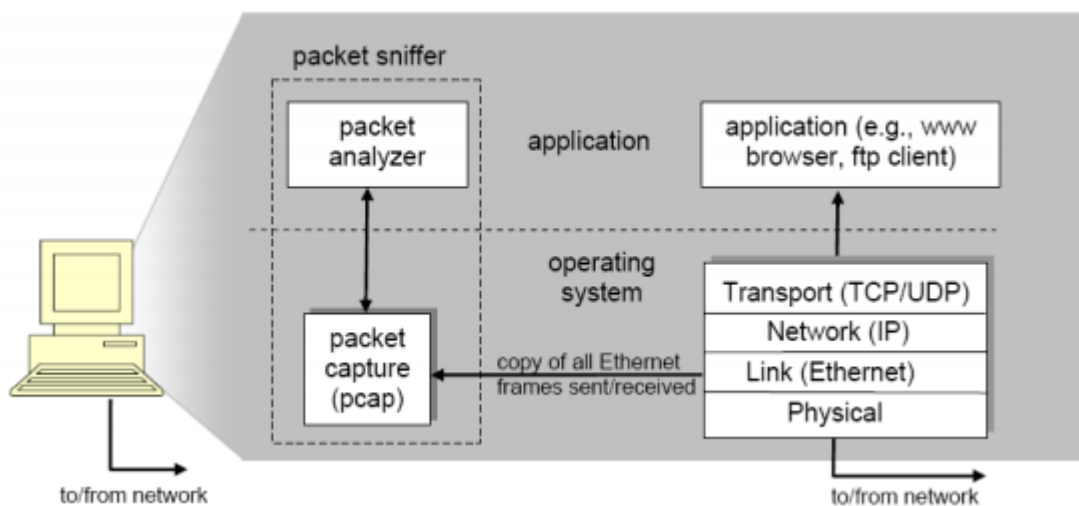


Netzwerkanalyse mit Wireshark

Das Monitoring inklusive der notwendigen Detailanalyse des Datenverkehrs im Netzwerk ist ohne ein leistungsfähiges Analysesystem unmöglich. Eines der **wichtigsten Netzwerktools** für jeden **Administrator** ist **Wireshark** – ein Open Source-Netzwerkanalysator, mit dem Sie alle **Pakete im Netzwerk aufzeichnen** und die Paketinhalte detailliert analysieren.

Packet Sniffer

Das grundlegende Werkzeug für die Beobachtung von Daten zwischen Rechnern wird als **packet sniffer** bezeichnet. Wie der Name schon sagt, **fängt** dieses Werkzeug **empfangene/gesendete Daten** Ihres Rechners **ab**. Ein Sniffer ist immer **passiv**, das heißt er **verschickt keine Daten**, sondern **speichert Kopien der Daten** der Kommunikation auf Ihrem Rechner.



Das Bild 1 zeigt die Struktur eines „Sniffers“. Rechts unten im Bild sind die beteiligten Protokolle abgebildet, es werden also Protokolle der Layer 1 bis 4:

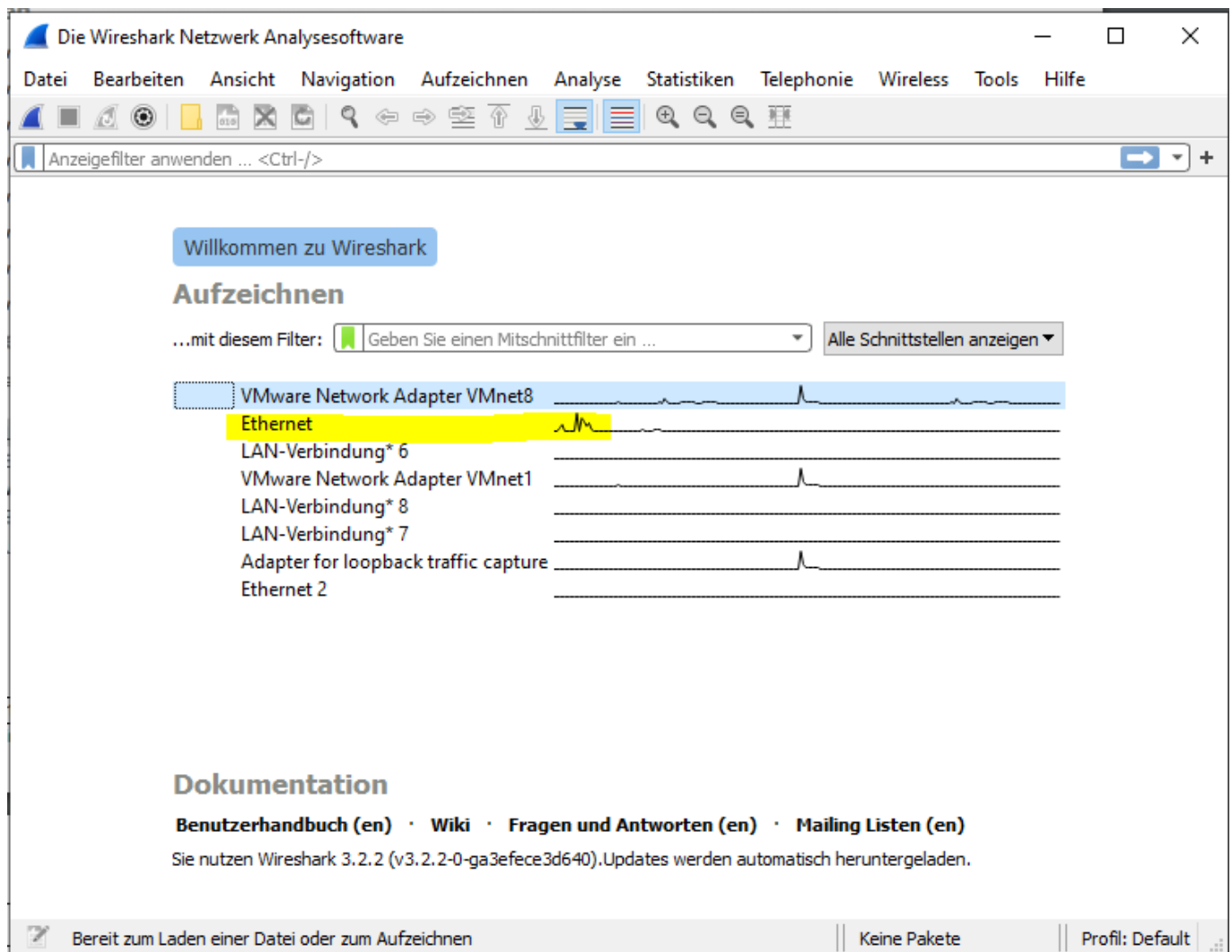
- oben ist die Applikation zu sehen, in unserem Fall ist dies der Webbrowser (Firefox, IE).
- die Blöcke die in den gestrichelten Linien eingerahmt sind gehören zu unserem „Sniffer“.

Am einfachsten lässt sich ein Netzwerk sniffen wenn Hubs als Verbindung der Netzwerkssegmente zwischengeschaltet sind. Bei anderen Verbindungen wie z.B. bei einem Switch bekommt man im Normalfall Probleme den Netzwerkverkehr abzuhören, da der Switch die Daten vom Sender nur an den tatsächlichen Empfänger weiterleitet. Somit würde man nur die eigenen Daten, sowie unwichtigeren Netzwerkverkehr wie z.B. Broadcasts aufzeichnen. Für diesen Fall haben viele Hersteller dieser Komponenten **Switches** bzw. **Router** mit einer **Monitorfunktion** in Ihrem Portfolio. Damit ist es möglich Netzwerkdaten eines gewünschten Ports auf einen anderen zu spiegeln. Auf diesen gespiegelten Port kann man nun direkt zugreifen.

Einführung in Wireshark

Startet man das Programm, muss man zuerst eine Schnittstelle (z.B.: Ethernet) wählen, die man

abhören möchte:



Der Bildschirm, in dem die aufgezeichneten Daten bearbeitet und analysiert werden ist in 3 Bereiche aufgeteilt:

1) Paketliste

In der Paketliste, sieht man alle aufgezeichneten Frames. Folgende Spalten werden standardmäßig angezeigt:

1. No. = ist eine fortlaufende Nummerierung der Frames
2. Time = zeigt den Zeitabschnitt der Aufzeichnung an
3. Source = zeigt den Absender eines Frames an (meist die IP)
4. Destination = zeigt den Empfänger des Frames an (meist die IP)
5. Protocol = zeigt das verwendete Protokoll des Frame an
6. Info = gibt zusätzliche Informationen zum Frame bekannt

Aufzeichnen von Ethernet

Datei Bearbeiten Ansicht Navigation Aufzeichnen Analyse Statistiken Telefonie Wireless Tools Hilfe

Anzeigefilter anwenden ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1158	214.841234	192.168.1.29	192.168.1.10	TCP	54	63336 → 443 [ACK] Seq=580
1159	215.590861	192.168.1.29	192.168.1.10	TLSv1.3	995	Application Data
1160	215.590942	192.168.1.29	192.168.1.10	TLSv1.3	155	Application Data
1161	215.591186	192.168.1.10	192.168.1.29	TCP	60	443 → 63336 [ACK] Seq=861
1162	215.984084	74.125.133.189	192.168.1.29	UDP	82	443 → 65264 Len=40
1163	216.010090	192.168.1.29	74.125.133.189	UDP	70	65264 → 443 Len=28
1164	216.202940	52.109.88.122	192.168.1.29	TLSv1.2	99	Application Data
1165	216.243318	192.168.1.29	52.109.88.122	TCP	54	63005 → 443 [ACK] Seq=71
1166	217.636990	192.168.1.29	185.199.109.153	TCP	55	[TCP Keep-Alive] 63228 →
1167	217.646662	185.199.109.153	192.168.1.29	TCP	66	[TCP Keep-Alive ACK] 443
1168	217.879712	108.177.15.189	192.168.1.29	UDP	82	443 → 59210 Len=40
1169	217.906999	192.168.1.29	108.177.15.189	UDP	70	59210 → 443 Len=28
1170	219.255630	192.168.1.20	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1
1171	219.782986	192.168.1.29	52.113.194.132	TCP	55	[TCP Keep-Alive] 63339 →
1172	219.794214	52.113.194.132	192.168.1.29	TCP	66	[TCP Keep-Alive ACK] 443
1173	220.790910	192.168.1.29	173.194.76.189	UDP	65	59222 → 443 Len=23
1174	220.830983	173.194.76.189	192.168.1.29	UDP	63	443 → 59222 Len=21

2) Paketdetails

In den Paketdetails werden die OSI-Layer (Schichten) des Datenframes angezeigt. Durch anklicken des Pfeil-Symbols kann der gewählte Layer erweitert werden.

```

> Frame 1384: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{428B0...
> Ethernet II, Src: Giga-Byt_4f:3a:d2 (00:1a:4d:4f:3a:d2), Dst: HewlettP_0b:2f:71 (24:be:05:0b:2f:71)
> Internet Protocol Version 4, Src: 192.168.1.10, Dst: 192.168.1.29
▼ Transmission Control Protocol, Src Port: 443, Dst Port: 63336, Seq: 10545, Ack: 8934, Len: 0
  Source Port: 443
  Destination Port: 63336
  [Stream index: 26]
  [TCP Segment Len: 0]
  Sequence number: 10545 (relative sequence number)
  Sequence number (raw): 1749144258
  [Next sequence number: 10545 (relative sequence number)]
  Acknowledgment number: 8934 (relative ack number)
  Acknowledgment number (raw): 2511940191
  0101 .... = Header Length: 20 bytes (5)
> Flags: 0x010 (ACK)
  Window size value: 434
  [Calculated window size: 55552]
  [Window size scaling factor: 128]
  Checksum: 0x366a [unverified]

```

Natürlich variieren die Protokolldetails von Protokoll zu Protokoll. Nur die ersten 2 Schichten sind immer vorhanden.

3) Paketdaten (Hexadezimal)

Hier sind die Daten einmal im Hexadezimalsystem (links) und nebendran im Klartext (rechts) bzw. in entschlüsselter Form angezeigt.

0000	24	be	05	0b	2f	71	00	1a	4d	4f	3a	d2	08	00	45	00	\$...	/q...	MO:...	E.
0010	00	28	8b	6c	40	00	40	06	2b	ec	c0	a8	01	0a	c0	a8	.	(.1@. @.	+
0020	01	1d	01	bb	f7	68	68	41	d2	c2	95	b9	2a	5f	50	10	hhA*	_P.
0030	01	b2	36	6a	00	00	00	00	00	00	00	00	00	00	00	00	..6j

Filter

Nachdem praktisch ständig Netzwerkpakete gesendet und empfangen werden, ist es wichtig, dass es Möglichkeit gibt, um den Netzwerktraffic zu filtern. In dem in der Abbildung gezeigten Textfeld können beliebige Filter eingestellt werden.

The screenshot shows the Wireshark network protocol analyzer interface. The filter bar at the top contains the filter: `http && ip.addr == 194.232.104.150`. The packet list shows two packets: packet 9256 (GET / HTTP/1.1) and packet 9262 (HTTP/1.1 301 Moved Permanently). The details pane for packet 9256 is expanded, showing the Hypertext Transfer Protocol section with fields like Host: `www.orf.at`, Connection: `keep-alive`, and User-Agent: `Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome...`.

Im obigen Beispiel wurden nur http-Pakete mit der IP-Adresse 194.232.104.150 angezeigt. Wie man in der Abbildung erkennen kann, ist/war dies die IP-Adresse der Internetseite www.orf.at. Da http natürlich das TCP-Protokoll, IP-Protokoll & Ethernet-Protokoll nutzt, werden auch diese Daten angezeigt.

From:
<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4:4_10

Last update: **2020/03/17 20:21**



Wireshark Übungen

1. Übung

1. Starte Wireshark
2. Starte einen Webbrowser mit einer beliebigen Seite
3. Filtere nach dem Protokoll http.
4. Öffnen die Seite <http://elearn.bgamstetten.ac.at>
5. Starte die Aufzeichnung nochmals neu und aktualisiere die Seite mit STRG+F5
6. Dokumentiere mittels Screenshot, wie viele gesendete und empfangen Pakete über die Schnittstelle mitgesniffen wurden.
7. Dokumentiere, welche IP-Adresse sich hinter dem FQDN elearn.bgamstetten.ac.at steckt.
8. Dokumentiere, welche Protokolle bei der ersten Antwort (text/html) zum Einsatz kamen.
9. Dokumentiere, wie groß die Antwort (text/html) in Bytes war.
10. Dokumentiere, wie groß
 - der Header des Ethernet 2 - Frames
 - der Header des IP-Paket-Pakets
 - der Header des TCP-Segments
 - und die Daten des http-Protokolls (gzip) der Antwort (text/html) waren.
11. Überprüfe, ob die die Größen zusammen die Gesamtgröße von zuvor ergeben.
12. Finde im Bereich Paketdetails den HTML-Quellcode und eine im Quellcode versteckte Botschaft. Der Name des input-Tags lautet STRENGGEHEIM und die Botschaft verbirgt sich value-Attribut.
13. Kontrolliere deine eigene MAC-Adresse (ipconfig /all) und finde die Ziel-MAC-Adresse heraus.
14. Finde heraus, ob IPv4 oder IPv6 als Protokoll auf der Netzwerkschicht verwendet wurde.
15. Finde heraus, welcher Port auf deinem PC bzw. am ELEARN-Server für die Netzwerkverbindung verwendet wurde.
16. Finde heraus, warum beim ersten Paket im TCP-Header die Sequence Number=1 ist und die Sequence Number beim übernächsten Paket nicht 2 sondern z.B.: 603 ist.
17. Finde heraus, welche HTTP-Version 1.0 oder 1.1 verwendet wurde.
18. Finde heraus, welcher User-Agent verwendet wurde.
19. Finde heraus, welche Sprachen vom Browser akzeptiert werden.
20. Bei der Antwort (text/html) wird ein HTTP-Status-Code mitgeschickt. Finde ihn heraus und recherchiere, was dieser Status Code bedeutet.
21. Finde heraus, welches Betriebssystem bzw. welcher Webserver verwendet wird.
22. Melde dich von der elearn-Plattform ab und anschließend wieder an. Finde heraus, ob und wie deine Login-Informationen mitgeschickt werden (Tipp Info=POST &do=login).

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4:4_10:4:4_10_01

Last update: **2020/03/17 21:31**



Wireshark Übungen

1. Übung

1. Starte Wireshark
2. Starte einen Webbrowser mit einer beliebigen Seite
3. Filtere nach dem Protokoll http.
4. Öffnen die Seite <http://elearn.bgamstetten.ac.at>
5. Starte die Aufzeichnung nochmals neu und aktualisiere die Seite mit STRG+F5
6. Dokumentiere mittels Screenshot, wie viele gesendete und empfangen Pakete über die Schnittstelle mitgesniff wurden.
7. Dokumentiere, welche IP-Adresse sich hinter dem FQDN elearn.bgamstetten.ac.at steckt.
8. Dokumentiere, welche Protokolle bei der ersten Antwort (text/html) zum Einsatz kamen.
9. Dokumentiere, wie groß die Antwort (text/html) in Bytes war.
10. Dokumentiere, wie groß
 - der Header des Ethernet 2 - Frames
 - der Header des IP-Paket-Pakets
 - der Header des TCP-Segments
 - und die Daten des http-Protokolls (gzip) der Antwort (text/html) waren.
11. Überprüfe, ob die die Größen zusammen die Gesamtgröße von zuvor ergeben.
12. Finde im Bereich Paketdetails den HTML-Quellcode und eine im Quellcode versteckte Botschaft. Der Name des input-Tags lautet STRENGGEHEIM und die Botschaft verbirgt sich value-Attribut.
13. Kontrolliere deine eigene MAC-Adresse (ipconfig /all) und finde die Ziel-MAC-Adresse heraus.
14. Finde heraus, ob IPv4 oder IPv6 als Protokoll auf der Netzwerkschicht verwendet wurde.
15. Finde heraus, welcher Port auf deinem PC bzw. am ELEARN-Server für die Netzwerkverbindung verwendet wurde.
16. Finde heraus, warum beim ersten Paket im TCP-Header die Sequence Number=1 ist und die Sequence Number beim übernächsten Paket nicht 2 sondern z.B.: 603 ist.
17. Finde heraus, welche HTTP-Version 1.0 oder 1.1 verwendet wurde.
18. Finde heraus, welcher User-Agent verwendet wurde.
19. Finde heraus, welche Sprachen vom Browser akzeptiert werden.
20. Bei der Antwort (text/html) wird ein HTTP-Status-Code mitgeschickt. Finde ihn heraus und recherchiere, was dieser Status Code bedeutet.
21. Finde heraus, welches Betriebssystem bzw. welcher Webserver verwendet wird.
22. Melde dich von der elearn-Plattform ab und anschließend wieder an. Finde heraus, ob und wie deine Login-Informationen mitgeschickt werden (Tipp Info=POST &do=login).

From:
<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:4:4_10:4_10_01

Last update: **2020/03/17 21:32**



Textverarbeitung & Zeichencodierung

Theorie

Textverarbeitung ist die Erstellung und Bearbeitung von schriftlichen Texten mit Hilfe von Programmen/Applikationen.

Bis weit ins 19. Jahrhundert war es üblich, Texte (sofern sie nicht gedruckt wurden) mit der Hand zu schreiben. Nachdem im 19. Jahrhundert die ersten praktisch verwendbaren Schreibmaschinen auf den Markt kamen, begann sich dies jedoch allmählich zu ändern. Nach und nach setzte sie sich gegenüber der Handschrift durch, ihre **Vorteile gegenüber der Handschrift**:

- **Klareres Schriftbild**
- **Normierte Zeichengrößen**
- **Schnelleres Schreiben**

Vor allem im Bürobereich fanden Schreibmaschinen deshalb große Verbreitung. Dies hatte **enormen Einfluss auf die Arbeitswelt**: Die bis dahin oft gebräuchlichen Stehpulte wurden durch Schreibtische abgelöst, **das Schreiben von einer stehenden zu einer sitzenden Tätigkeit**. Dieser Prozess lief parallel zu anderen Entwicklungen in der Mechanisierung der Büroarbeit (Einführung von Rechenmaschinen, Hollerithmaschine und Telefon).

Mit der zunehmenden Verbreitung von PCs in den 1990er Jahren verschwanden Schreibmaschinen jedoch vom Markt.

PCs waren **universell einsetzbar** und **nicht nur auf Textverarbeitung** beschränkt, bei zunehmend günstigerem Preis-Leistungs-Verhältnis zugunsten des PCs. Die Fähigkeiten von Textverarbeitungen erforderten zunehmend vollwertige Computer, vor allem nach dem Aufkommen grafischer Benutzeroberflächen. Seit **Einführung des Personal Computers** hat sich der **Bereich der Textverarbeitung rasant entwickelt**. Das heutige Ergebnis sind Programme, bei denen der reine Textverarbeitungsteil programmiertechnisch wohl noch den geringsten Aufwand erforderte. Vielmehr wurden sie angereichert mit Funktionen, die zuvor einzeln von anderen Programmen und oft auch anderen Herstellern bezogen werden mussten. **Heutige Textverarbeitungsprogramme** integrieren die reine Texterfassung mit den **Möglichkeiten der Grafikeinbindung, Tabellenerstellung, Formelgenerierung oder der Präsentationsgrafik**.

Die Einführung der grafischen Benutzeroberflächen wie die des Apple Macintosh 1984 und später Windows ermöglichte erst den Schritt in diese Richtung, verhalf sie doch den Programmen, auf Techniken wie dem **Datenaustausch über eine zentrale Zwischenablage**, dem dynamischen Datenaustausch oder dem Objekt-verbinden-und-einbetten zurückzugreifen. Die Möglichkeit, den Text gemäß dem Prinzip **WYSIWYG (engl. What You See Is What You Get)** bereits auf dem Bildschirm so präsentiert zu bekommen, wie er auch später ausgedruckt erscheint (Druckdarstellung), war dabei von großer Bedeutung.

WYSIWYG

WYSIWYG ist das Akronym für den Grundgedanken **What You See Is What You Get** – auch als **Echtzeitdarstellung** bekannt (beziehungsweise, bezogen auf die Bildschirmdarstellung,

Echtbilddarstellung). Bei echtem WYSIWYG wird ein Dokument während der Bearbeitung am Bildschirm genauso angezeigt, wie es bei der Ausgabe über ein anderes Gerät, z. B. einen Drucker, aussieht. Der Begriff wurde zuerst Anfang der 1980er im Zusammenhang mit Computer-Drucksatz-Systemen (Desktoppublishing) und Textverarbeitungsprogrammen verwendet.

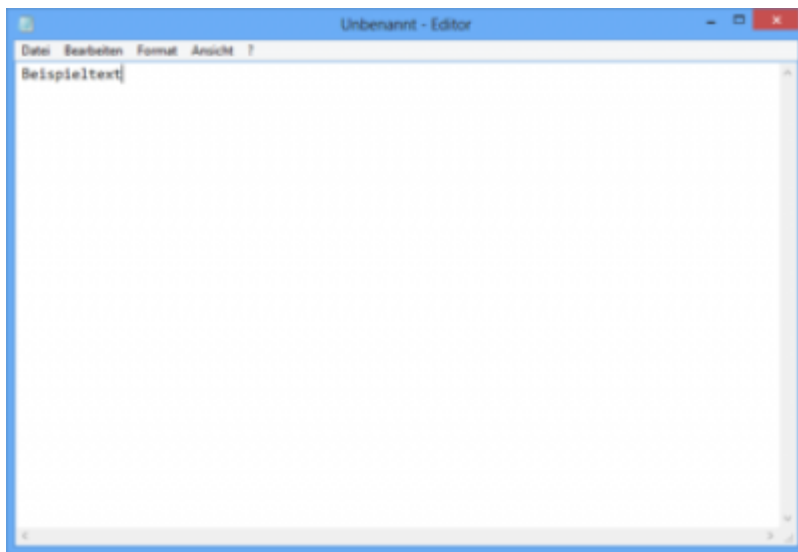
Texteditoren und ASCII-Editoren

Die ersten **Textverarbeitungsprogramme**, treffender bezeichnet als „Texterfassungsprogramme“, waren **einfache Texteditoren**, die sich aus dem **Zeichenvorrat nichtgenormter Zeichensätze** bedienten. Somit waren nicht nur die **Editoren an den zugehörigen Computer (bzw. das zugehörige Betriebssystem) gebunden**, sondern auch die darauf erstellten Texte, die bei Bedarf erst für andere Computersysteme in andere Zeichensätze konvertiert werden mussten.

Erst mit der Einführung **genormter Zeichensätze (z.B. ASCII)** kann man von **ASCII-Editoren** sprechen, die sich aus dem Zeichenvorrat der **standardisierten ASCII-Tabelle** bedienen. Damit wurden auch die Editorprogramme selbst grundlegend **kompatibel zu gleichartiger Hardware unter den meisten gängigen Betriebssystemen**.

Der Befehlsumfang einfacher ASCII-Editoren ging in der Regel über das Speichern, Laden und Drucken nicht hinaus.

Ein Beispiel für einen reinen ASCII-Editor ist der Editor unter Microsoft Windows.



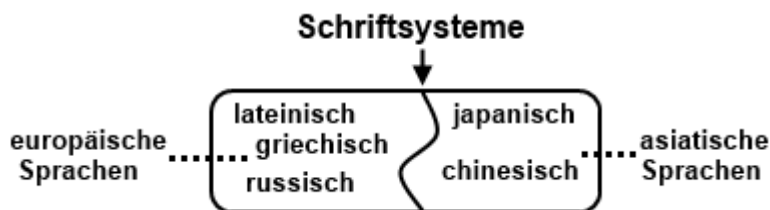
Erweiterte Programme bieten jedoch neben Funktionen wie Textmarkierung oder das Ausschneiden, Kopieren und Versetzen von Textabschnitten an: **Syntaxhervorhebung und Makroprogrammierung**. Besonders **komplexe Editoren** wie **Emacs und Vim** unter Linux kommen einer modernen Textverarbeitung unter den Einschränkungen eines textorientierten Betriebssystems bereits sehr nahe und berücksichtigen die Bedürfnisse von Systemadministratoren oder Programmierern, etwa durch **Verknüpfung mit Compilern** und anderer **externer Software**.


```
geek@ubuntu: ~  
File Edit View Search Terminal Help  
  
~  
~  
~  
VIM - Vi IMproved  
~  
version 8.0.1453  
by Bram Moolenaar et al.  
Modified by pkg-vim-maintainers@lists.alioth.debian.org  
Vim is open source and freely distributable  
  
Help poor children in Uganda!  
type :help iccf<Enter>      for information
```

Zeichencodierung

Da in der frühen Phase der **Computertechnologie** die ersten Computersysteme in den **westlichen Ländern** entwickelt wurden, vor allem in den USA, **konzentrierten** sich die **Entwickler von Textverarbeitungsprogrammen** zunächst auf die Kodierung von Texten, die auf dem **lateinischen Schriftsystem** basierten.

- Eine **Schrift** besteht aus einer **bestimmten Anzahl von Schriftzeichen**.
- Diese **Anzahl unterscheidet sich stark**, je nach verwendetem **Schriftsystem**



Während Schriftsysteme **europäischer Sprachen**, wie z.B. lateinisch, griechisch und russisch aus einer relativ geringen Anzahl von **Schriftzeichen** bestehen, enthalten die **Schriftsysteme asiatischer Sprachen**, wie z.B. chinesisch und japanisch eine **wesentlich größere Anzahl** von Schriftzeichen.

Bits, Bytes und Buchstaben

Computersysteme kennen im Prinzip **keine Buchstaben** oder **Zeichen**, sondern nur **Bits und Bytes**.

⇒ daher muss bei der Textverarbeitung auf Computersystemen die **Schriftzeichen** erst in Bits (0 und 1) **umgewandelt („kodierte“)** werden! Dies macht der Computer nach einer **Zuordnungstabelle**, in der verzeichnet ist, welche Zahl für welchen Buchstaben steht. Diese Tabelle ist in beiden Richtungen eindeutig, aus einem Buchstaben wird also exakt eine Zahl und aus einer Zahl wird umgekehrt wieder exakt dieser Buchstabe!

Drückt man also eine Taste der Tastatur, erkennt der Computer, welche Taste gedrückt wurde, und ordnet ihr aufgrund des konfigurierten Tastaturlayouts den Bytewert des Buchstabens zu, der sich dahinter verbirgt. Bei der Ausgabe des Buchstabens auf dem Bildschirm wird der Bytewert umgewandelt in die Anweisung, wie der entsprechenden Buchstabe zu malen ist, und er erscheint dann auf dem Bildschirm. Wie der Buchstabe konkret aussieht, entscheidet dabei die **Malanweisung der Schriftart**. **Ändert man die Schriftart**, wird eine **neue Malanweisung** verwendet, aber die **Bytwerte der Zeichen bleiben identisch**.

Diese betagten Codes wie ASCII und EBCDIC haben jedoch eine Schwäche: Mit ihnen lassen sich, da sie einem Buchstaben ein Byte zuordnen, **maximal 256 verschiedene Zeichen** darstellen. Dieser Zeichenvorrat **reicht gerade für das lateinische Alphabet** und **einige zusätzliche Zeichen**. Als die Codes in den **1960ern** entstanden, war das für die englisch sprechenden Amerikaner **ausreichend**. Als ASCII entstand, kam man – zumindest in der Datenverarbeitung im englischen Sprachraum – mit 128 verschiedenen Zeichen aus, es reichten daher **7 Bit für die Codierung des Zeichensatzes**. Das **8. Bit** eines Bytes wurde **nicht benutzt** oder für verschiedene andere Zwecke benutzt!

Die erste, nicht hundertprozentig befriedigende Lösung für dieses Problem bestand darin, für andere Alphabete neue Zuordnungen zu erstellen. Dies bedeutete bei der interkulturellen Kommunikation aber einen kleinen Mehraufwand. Wollte beispielsweise ein Russe einem Amerikaner einen russischen Text zukommen lassen, so reichte es nicht, einfach die Textdatei weiterzugeben. Der Absender musste zusätzlich die für das kyrillische Alphabet verwendete Codierung angeben. Mit der amerikanischen Codierung würde der Computer des Empfängers die Bytes als lateinische Buchstaben interpretieren und somit einen sinnlosen Buchstabensalat produzieren.

ISO-8859-Familie

Im **westeuropäischen Sprachraum** verbreiteten sich insbesondere **zwei Codierungen aus der sogenannten ISO-8859-Familie**:

- ISO-8859-1 und
- ISO-8859-15.

Den Codierungen der ISO-8859-Familie ist gemeinsam, dass sie nur 256 verschiedene Zeichen codieren können, da ein Zeichen immer mit einem Byte codiert wird. Jede dieser Codetabellen versucht, möglichst alle Zeichen möglichst vieler Schriften zu speichern. Sie **verwenden alle die ASCII-Codetabelle für die Zeichen 0 bis 127** und definieren **zusätzlich die Werte 160 bis 255**. Sofern in diesem Bereich das gleiche Zeichen in verschiedenen Schriften vorkommt, überlappen sie sich zumeist.

Upper part of ISO 8859-1 character set

160 NBSP	161 ı	162 Œ	163 £	164 ¤	165 ¥	166 ¦	167 §	168 ¨	169 ©	170 ª	171 «
172 ¬	173 Š	174 Œ	175 —	176 °	177 ±	178 ²	179 ³	180 ´	181 µ	182 ¶	183 ·
184 ¸	185 ı	186 ²	187 »	188 ¼	189 ½	190 ¾	191 ¿	192 À	193 Á	194 Â	195 Ã
196 Ä	197 Å	198 Æ	199 Ç	200 È	201 É	202 Ê	203 Ë	204 Ì	205 Í	206 Î	207 Ï
208 Ð	209 Ñ	210 Ò	211 Ó	212 Ô	213 Õ	214 Ö	215 ×	216 Ø	217 Ù	218 Ú	219 Û
220 Ü	221 Ý	222 Þ	223 ß	224 à	225 á	226 â	227 ã	228 ä	229 å	230 æ	231 ç
232 è	233 é	234 ê	235 ë	236 ì	237 í	238 î	239 ï	240 ð	241 ñ	242 ò	243 ó
244 ô	245 õ	246 ö	247 ÷	248 ø	249 ù	250 ú	251 û	252 ü	253 ý	254 þ	255 ÿ

Der universale Code: Unicode

Mit **Unicode** werden so ziemlich **alle Zeichencodierungsprobleme** dieser Welt **gelöst**. Und weil Webseiten potentiell mit allen Schriften dieser Welt genutzt werden, ist für HTML 4.0 und XML (und damit auch XHTML) festgelegt, dass grundsätzlich alle in Unicode definierten Zeichen vorkommen dürfen.

Mit dieser Festlegung kann man also jedes der mittlerweile über **100.000 Unicode-Zeichen** in seinem Text verwenden. Aber Unicode regelt nicht nur die Codierung aller Schriftzeichen dieser Welt, sondern kennt zu jedem Zeichen dutzende definierte Eigenschaften. Ein Zeichen besitzt mindestens folgende Informationen:

- der eindeutige Unicode-Name des Zeichens
- den zugeordneten Codepoint (die „Nummer“ des Zeichens)
- eine generelle Kategorisierung (Buchstabe, Zahl, Symbol, Interpunktion, ...)
- weitere grundsätzliche Charakteristiken (Whitespace, Bindestrich, alphabetisch, ideographisch, Nicht-Zeichen, veraltet, ...)
- anzeigerelevante Informationen (Schreibrichtung, Form, Spiegelung, Breite)
- Groß-/Kleinschreibung/Zahlenwert und -typ bei Zahlzeichen
- Trennfunktionen (zwischen Wörtern, Zeilen, Sätzen,...)
- usw.

Die meisten Eigenschaften eines Zeichens sind nur für die interne Behandlung durch den Computer relevant. Beispielsweise werden arabische Schriftzeichen automatisch korrekt von rechts nach links geschrieben, oder der Zeilenumbruch nutzt die Stellen, an denen eine Wortgrenze z. B. durch ein normales Leerzeichen existiert. Dieser kleine Exkurs soll aber verdeutlichen, dass die Bytewerte der Buchstaben für den Computer viel mehr bedeuten können, als nur den Buchstaben an sich.

Der oben erwähnte Codepoint eines Unicode-Zeichens ist nur **eine abstrakte Nummer**. Die Schreibweise dieser Nummer im Unicode-Standard erfolgt in **hexadezimalen Zahlen mit vorangestelltem „U+“**. Der Codepoint legt noch keinerlei computerkompatible Darstellung fest, dies ist Aufgabe des Codierschemas. Da die Unicode-Codepoints von U+0000 bis U+10FFFF (hexadezimale Zahlendarstellung), mit einer beabsichtigten Lücke zwischen U+D7FF und U+E000, reichen, sind für eine vollständige Codierung des gesamten Codepoint-Bereichs als Binärzahl mindestens 3 Byte erforderlich.

Folgende Speichervarianten von Unicode sind heute gängig:

- **UTF-8** (8 Bits \Rightarrow 1 Byte)
- **UTF-16** (16 Bits \Rightarrow 2 Byte)
- **UTF-32** (32 Bits \Rightarrow 3 Byte)

Praxistipp

Empfehlung: Verwenden Sie, wo immer es geht, UTF-8! Es hat viele Vorteile, bei der Wahl der codierbaren Zeichen nicht eingeschränkt zu sein. Selbst wenn ein Webauftritt garantiert nur für eine einzige Sprache (z. B. Deutsch) gestaltet werden muss, bietet nur Unicode die komplette Freiheit, wirklich alle Zeichen – insbesondere Interpunktionszeichen wie „typografische Anführungszeichen“ oder eben das Eurozeichen – beliebig verwenden zu können. Eingestreute Fremdwörter in einer anderen Sprache sind genauso problemlos möglich wie die Verwendung eher exotischerer Zeichen wie den Brüchen $\frac{1}{3}$, $\frac{2}{3}$, $\frac{1}{8}$, $\frac{3}{8}$, $\frac{5}{8}$, $\frac{7}{8}$.

Viele Programme können schon seit langer Zeit mit UTF-8 umgehen: Server, Browser, Datenbanken, Programmiersprachen, Editoren. Der einzige Grund, auf UTF-8 zu verzichten, ist die Existenz von alten Systemen oder Daten, die nicht in UTF-8 gespeichert sind und bei denen es zu aufwendig wäre, sie nach UTF-8 zu konvertieren.

Begrifflichkeiten

Für die Zeichen-Problematik gibt es eine Reihe Begrifflichkeiten, die allerdings oftmals falsch angewendet werden. Nachfolgend der Versuch einer Aufklärung:

Zeichensatz

Ein **Zeichensatz (character set, kurz charset)** ist die **Gesamtheit der zur Verfügung stehenden Zeichen**. Ein Zeichensatz ist ein eher abstraktes Gebilde, das nur die Zeichen selbst und eine Reihenfolge beschreibt, **nicht jedoch eine konkrete Abbildung auf Byte-Werte**. Das ist Aufgabe der **Zeichencodierung**.

Beispielsweise ist **Unicode ein Zeichensatz**, **UTF-8** hingegen ist eine **Zeichencodierung**.

Codepoint

Zeichen in einem Zeichensatz (z. B. ASCII oder Unicode) werden in einer **bestimmten Reihenfolge in einem Coderaum (code space)** angeordnet. Die Position eines Zeichens ist der Codepoint (zu Deutsch etwa „**Codeposition**“).

Das heißt die Codeposition ist der Zahlenwert eines Zeichens im Coderaum. \

Bsp.:

A hat die Codeposition 65 im Coderaum ASCII.

Ein **Zeichensatz mit Codepoints** ist ein **codierter Zeichensatz (coded character set)**.

Schriftart

Für die grafische Darstellung von Zeichen wird eine Schriftart verwendet, die in der Regel eine in sich **konsistente Gestaltung** aufweist, beispielsweise für die **Strich-Dicke** oder **Verzierungen**. Arial, Times New Roman oder Courier sind Beispiele für Schriftarten.

Schriftart ist im engeren Sinne nach eher ein **Synonym zu Schriftschnitt**, also bspw. **Helvetica 24pt fett**. Da Schriftarten wie Arial, Helvetica, Times Roman usw. oft aus mehreren Schriftschnitten für Breiten (schmal, breit, ...), Strichstärken (leicht, normal, fett, ...) und Zeichenlagen (normal, kursiv, ...) bestehen, verwendet man auch den Begriff **Schriftfamilie**, der nicht genau von **Schriftart** **abzugrenzen** ist.

Vanessa - Ihr Wunschtext - 0123456789

SL - Ihr Wunschtext - 0123456789

Arndt - Ihr Wunschtext - 0123456789

Champagne - Ihr Wunschtext - 0123456789

Edwardian - Ihr Wunschtext - 0123456789

Nuance - Ihr Wunschtext - 0123456789

Comic - Ihr Wunschtext - 0123456789

Harrington - Ihr Wunschtext - 0123456789

Mono - Ihr Wunschtext - 0123456789

Trianti - Ihr Wunschtext - 0123456789

Times - Ihr Wunschtext - 0123456789

Englisch - Ihr Wunschtext - 0123456789

Twiles - Ihr Wunschtext - 0123456789

Kriterien für Schriftarten

Die Kriterien für die Verwendung einer bestimmten Schriftart können sein:

- **Verfügbarkeit:** Die Windows-Schriftarten wie Arial zum Beispiel sind gerade deshalb weit verbreitet, weil sie auf jedem Windows-System verfügbar sind und entsprechend formatierte Dokumente auch unter mehreren Benutzern leicht ausgetauscht werden können
- **Emotionale Wirkung:** Ein Text kann die unterschiedlichsten Zwecke erfüllen. Entsprechend seiner Aufgabe wird er in einer dazu passenden Schrift gesetzt. Im Verkehrsbereich finden vorwiegend Grotesken, wie die Univers, Anwendung, in belletristischer Literatur wird gerne auf Renaissance-Antiqua wie die Garamond zurückgegriffen, und eine spielerischere Schriftart ist die Comic Sans MS, die auch als „Kinder-Schriftart“ benutzt wird.
- **Lesbarkeit:** Sie ist wichtig, um ein Buch entspannt lesen zu können oder weiter entfernte Straßenschilder zu entziffern. Das Auge des Lesers muss möglichst leicht die einzelnen Buchstaben eines Textes voneinander trennen und erkennen können. Einfluss auf die Lesbarkeit einer Schrift hat neben der Gestaltung der Wort- und Zeichenzwischenräume auch die Gestalt der Zeichen selbst.

Schriftgestalt

Serifen

Serifen sind **kleine Endstriche eines Buchstabens**, umgangssprachlich auch „**Füßchen**“ genannt. Sie bilden eine horizontale Linie, an der sich das **Auge des Lesers orientieren** kann. Daher eignen sich Serifenschriften, insbesondere die Antiqua, **besonders für gedruckten Fließtext (Bücher,**

Artikel). Bei Postern, Plakaten, Schildern usw. kommt es dagegen darauf an, auch auf größere Distanz einzelne Wörter zu entziffern. Hier werden wegen ihrer größeren Klarheit Schriften ohne Serifen eingesetzt.

Courier: Wi wi

Times: Wi wi

Arial: Wi wi

Verdana: Wi wi

Höhe

Neben der Tatsache, dass es zwei parallele Alphabete von Groß- und Kleinbuchstaben gibt, unterscheiden sich seine Buchstaben noch in weiteren Punkten. Zum Beispiel betrachtet man die Höhe der einzelnen Buchstaben. Zunächst einmal allen gemeinsam ist, dass sie eine Grundlinie teilen. Die **Grundlinie ist der Boden der Zeile** selbst, der bei Serifenschriften durch ihre Füße angedeutet wird. Auf dieser Grundlinie „stehen“ alle Buchstaben und gehen unterschiedlich weit über bzw. unter diese hinaus.



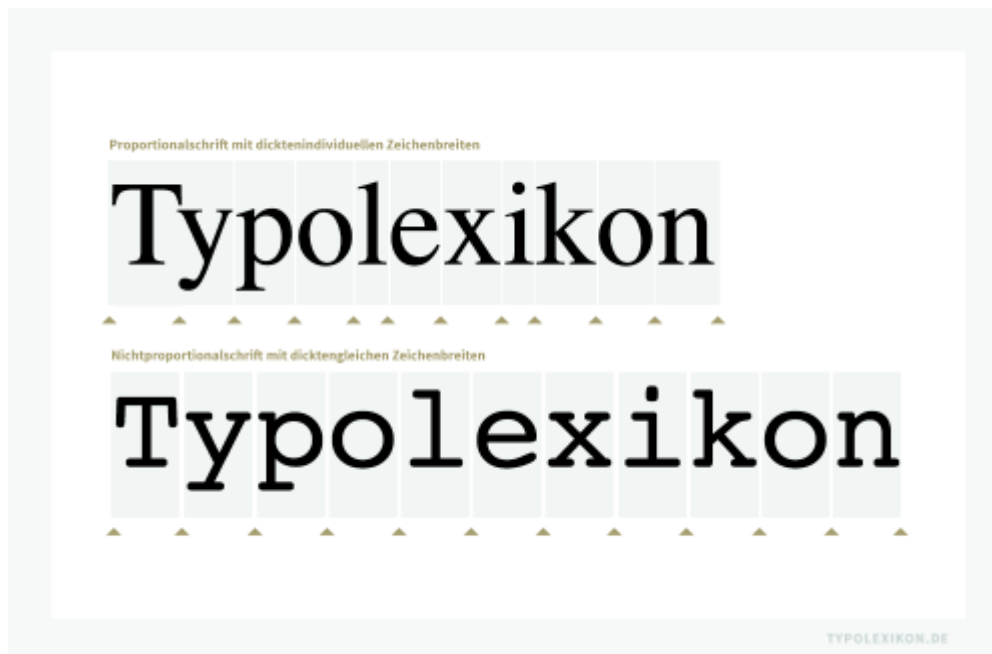
Laufweite

Die **Laufweite** einer Schrift beschreibt, wie groß die **horizontale Ausdehnung eines geschriebenen Textes** ist. Sie entsteht durch die Breite der einzelnen Buchstaben sowie den Abstand, den sie zueinander haben. Die Laufweite spielt **beim Schriftsetzen in Büchern, Zeitungen, Magazinen** usw. eine große Rolle, da sie bestimmt, wie „**ökonomisch**“ eine Schrift ist, das heißt, wie **platzsparend** sie ist. Um wertvollen Platz zu sparen, gibt es gerade für diesen Zweck entworfene Schmal-Schnitte einer Schrift (englisch „condensed“). Wesentlich seltener werden extra breite Schnitte verwendet (englisch „extended“), die meisten hiervon sind Grotesken (Sans Serifs). Der **Verwendungszweck von breiten Schnitten** ist weniger ökonomisch als **grafisch (für Überschriften, Plakate, Logos, Corporate Design usw.)**.

so close and yet so far away
so close and yet so far away
so close and yet so far away
so close and yet so far away

Proportionen

Im **Normalfall** sind die einzelnen **Zeichen einer Schriftart unterschiedlich breit**, ein „w“ nimmt also mehr Platz ein als ein „i“. Solche Schriftarten werden **proportional** genannt. Verbreitete Mitglieder dieser Gruppe sind zum Beispiel **Times**, **Arial** oder **Lucida**. Um die **Konstruktion von Schreibmaschinen und Computern zu vereinfachen**, kamen **nichtproportionale**, sogenannte **dicktengleiche Schriftarten zum Einsatz**, bei denen alle Zeichen eine identische Breite aufweisen. Die wohl bekanntesten dieser Schriften sind **Courier** und **Lucida Console**.



Glyphe

Der Begriff **Glyphe** wird gelegentlich für die **konkrete Darstellung eines Zeichens** verwendet. Die Schriftart bestimmt, welches Zeichen mit welcher Glyphe dargestellt wird.



Font

Ein **Font** ist die **Aufbereitung einer Schriftart für den Einsatz mit einem Computer**, also eine **Datei**, die eine **Schriftart beschreibt**.

Tastatur-Layout

Ein **Tastatur-Layout** oder eine **Tastaturbelegung** (auch Tastenlayout oder Tastenbelegung) **ordnet** einer Taste auf der **Tastatur einen Codepoint** zu.

Beispielsweise produziert die auf einer deutschen Tastatur mit „z“ beschriftete Taste mit einer deutschen Tastenbelegung den Unicode-Codepoint U+007A, also (erwartungsgemäß) ein „z“. Ändert man die Tastenbelegung auf US-amerikanisch, so erzeugt dieselbe Taste den Codepoint U+0079, also ein „y“. Mit einer russischen Belegung produzieren praktisch sämtliche Tasten andere Resultate, nämlich die Codepoints für kyrillische statt lateinischer Buchstaben, also z.B. „н“ (U+043D) statt „z“.

Textverarbeitungsprogramme

Textverarbeitungsprogramme bieten im Gegensatz zu reinen Texteditoren in der Regel **erweiterte Layout- und Formatierungsfunktionen** an. Neben der Textüberarbeitung erhöhen **Rechtschreibprüfung, Indexerstellung, Such- und Ersetzfunktionen** den Nutzen für den Anwender. **Formatvorlagen** vereinfachen zudem eine einheitliche Gestaltung der zu veröffentlichenden Dateien, Textbausteine die Einbindung von wiederkehrenden Inhalten.

Zeichenorientierte Textverarbeitung

Die Kategorie der im Funktionsumfang erweiterten zeichenorientierten Textprogramme (**Character Used Interface**) verwendet ebenfalls, wie die ASCII-Editoren auch, den normierten und **beschränkten ASCII-Zeichensatz** als Grundlage. Die fest definierten Zeichensätze werden sowohl für die Bildschirmdarstellung als auch für den Ausdruck verwendet. Der **Abstand der Zeichen untereinander ist fest vorgegeben**, wie auch die Zeichen selbst statisch sind. Diese „**Statik**“ bedeutet, dass alle **Zeichen des Zeichenvorrats vorgeformt** und **fertig zur Verfügung** stehen. Hier liegt der **große Vorteil der CUI-Programme** begründet: **die Arbeitsgeschwindigkeit**. Da alle **Zeichen in Größe und Form fertig** vorliegen, bedarf es **keiner ständigen**

Neuberechnung von deren Bildschirmdarstellung. Die **Hardware-Anforderungen** beim Einsatz von CUI-Programmen sind entsprechend **gering**. Der dadurch erkaufte Nachteil ist jedoch die heute gewünschte Darstellungsqualität, die fortwährende Ansicht als Seitenvorschau.

Grafisch orientierte Textverarbeitung

Die grafisch orientierten Programme basieren auf dem **GUI, dem Graphical User Interface**. Alle **Zeichen** sind in **Form und Größe variabel**. Am Bildschirm können also **Formatierungen und Schriftgrößen exakt so angezeigt werden**, wie sie formatiert wurden und später auch auf dem **Ausdruck** erscheinen. Ermöglicht wird dies durch die **getrennte Handhabung von Schriftzeichen für den späteren Ausdruck** zum einen und **Bildschirmschriften** zum anderen. Der Preis hierfür ist jedoch ein **enormer Rechenaufwand** für die ständige **Aktualisierung und Neuberechnung der Bildschirmanzeige**. **Wichtig** geworden sind deshalb neben der reinen **Prozessorleistung** des Rechners Komponenten wie die **Grafikkarte** und der Bildschirmspeicher. Das Druckergebnis soll also exakt dem entsprechen, was auf dem Bildschirm zu sehen ist (**WYSIWYG, What You See Is What You Get**). Oder anders ausgedrückt: **Alles, was und wie man es auf dem Bildschirm sieht, soll auch genauso ausgedruckt werden**.

Dateiformate

Jedes **Textverarbeitungsprogramm** hat bis jetzt noch **sein eigenes Dateiformat** für die Speicherung von Dokumenten. Eine gewisse **Standardisierung** ist durch das **Rich Text Format** erreicht worden, mit dem aber die **Einheitlichkeit des Layouts auf verschiedenen Rechnern nicht gewährleistet** ist.

Eines der am weitesten verbreiteten Dokumentenformate zum Austausch von Dokumenten, die nicht zur weiteren Bearbeitung vorgesehen sind, ist **PDF**. Dieses wird als **ergänzendes Standardformat** von immer mehr Textverarbeitungsprogrammen oder **als separater virtueller Druckertreiber als Exportfunktion** (d. h. zum Speichern) angeboten, kann unter den meisten gängigen Betriebssystemen (darunter auch einige ältere Versionen) eingelesen werden und **behält dabei das Aussehen**.

Nachteil: Man kann das **PDF-Dokument nicht mehr bearbeiten**.

Klassifikation von Textverarbeitungsprogrammen

Textverarbeitungsprogramme können nach unterschiedlichen Kriterien voneinander unterschieden und klassifiziert werden:

- Komplexität (vom einfachen Texteditor bis zur hochkomplexen Office-Software)
- Verwendungszweck (Textverarbeitungsprogramme für Briefe, Verfassen von Quellcode z.B. notepad++)
- Plattformunterstützung (Linux, Windows, iOS, Android, ...)
- Lizenzierung (Freeware oder Commercial)

Kritik

Das Verfassen am Bildschirm könne dazu führen, dass man einen **geringen Überblick** über den Text habe.

Texte würden **nicht mehr selbst erstellt und durchdrungen (und auch kaum noch gelesen)**, sondern mittels **Kopieren und Einfügen** zusammengesetzt. Dieses Phänomen zeige sich besonders bei **ergoogelten Plagiaten**

Praxis mit MS Word

<https://support.office.com/de-de/word>

Folgende Befehle/Funktionen sollte dir in MS Word geläufig sein! Falls nicht erkunde die Hilfe von MS Word.

- Absatz/Zeilenende/Seitenumbruch
- Lineal
- Absatzzeichen
- Absätze
- Einzüge
- Tabstopps
- Ansichten
- Silbentrennung
- Suchen & Ersetzen
- Seite einrichten
- Zeilenabstandsoptionen
- Formatvorlagen
- Aufzählungen/Nummerierungen
- Tabellen
- Bilder
- Symbole
- Rahmen
- Wasserzeichen
- Serienbrief mit Adressen
- Seitenumbruch
- Kopf und Fußzeile
- Seitenzahl (erste Seite anders)
- Format übertragen
- Rechtschreibung
- Überprüfen
- Vorlage mit Formularen
- Zitate
- Inhaltsverzeichnis
- Abbildungsverzeichnis
- Literaturverzeichnis

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:5



Last update: **2020/04/20 09:19**

Tabellenkalkulation

Theorie

Eine Tabellenkalkulation ist eine Software für die interaktive Eingabe und Verarbeitung von numerischen und alphanumerischen Daten in Form einer Tabelle. Vielfach erlaubt sie zusätzlich die grafische Darstellung der Ergebnisse in verschiedenen Anzeigeformen.

Das Bildschirmfenster der Software ist dabei in Zeilen und Spalten eingeteilt. Je nach Programm bzw. Bedienungskonzept heißt dieser Bereich zum Beispiel Arbeitsblatt, Worksheet oder Spreadsheet. Jede Zelle der Tabelle kann eine Konstante (Zahl, Text, Datum, Uhrzeit ...) oder eine Formel enthalten. Für die Formeln stehen meist zahlreiche Bibliotheksfunktionen zur Verfügung. Die Formeln können Werte aus anderen Zellen benutzen.

Bei Änderung der referenzierten Zellen einer Formel aktualisiert die Software den erst angezeigten Wert der Formelzelle normalerweise automatisch, ggf. aber auch nur auf Anforderung. Werden Formeln eines Tabellenfeldes an andere Stellen kopiert, muss zwischen absolutem und relativem Zellbezug unterschieden werden. Formelzellen können auf andere Formelzellen verweisen. Mit diesem Prinzip können komplizierte Rechengänge mit vielen verknüpften Teil-Ergebnissen übersichtlich dargestellt werden.

Wozu?

Die Erfindung der elektronischen Tabellen zusammen mit Textverarbeitungs-Software und Datenbanken war zweifellos ein wesentlicher Faktor im Menschen den Wert von Mikrocomputern in den Anfangsjahren nach Apple und IBM begann Vermarktung Personalcomputer zu überzeugen. Seit dieser Zeit haben die stetig wachsende Vielseitigkeit und weiteren Anwendungen der Tabellenkalkulations-Software es in ein Produkt gemacht, die scheint fast unentbehrlich für Unternehmen und Privatanwender. Tabellen sind jetzt ein Standardbestandteil der Office-Suite-Pakete.

Funktionen

Kalkulationstabellen können mehr als einfache arithmetische Berechnungen durchführen. Eine Tabellenkalkulation übersetzen komplizierte Daten und Berichte in eine Kombination von Zahlen und Grafiken. Moderne Versionen enthalten eine umfangreiche Liste von finanzielle Rechner, z. B. Interesse Berechnungen, Kreditberechnung, auch Berechnungen für Schatzwechsel Preise. Statistische Funktionen von üblichen Berechnungen (Chi-Quadrat, Pearson-Koeffizient der Korrelation und Standardabweichung) auf abstruse Funktionen wie die hypergeometrische Verteilung und die Poisson-Verteilung Rückgabewerte benötigten mit keinen Schmerz. Es gibt mindestens 100 dieser Formeln in zeitgenössischen Tabellen enthalten.

Tabellen können als einfachen Datenbanken funktionieren. Durch das Einfügen von Daten und Anzahl in verschiedenen Spalten, können die Ergebnisse sortieren, gesucht oder gefiltert. Was-wäre-wenn-Analyse läuft, wenn Daten aus Zellen in einem Arbeitsblatt. Es gibt auch mathematische und trigonometrische Funktionen. Ein Benutzer kann eine vereinfachte Mailing-Liste in einer

Kalkulationstabelle erstellen, durch die Eingabe von Namen und Adressen in einzelnen Spalten. Es gibt sogar Funktionen der Textverarbeitung bietet Kontrolle über Schriftarten, Fett oder kursiv-Schrift, Größe, Farbe und Seitenformatierung.

Bei Änderung der referenzierten Zellen einer Formel aktualisiert die Software den erst angezeigten Wert der Formelzelle normalerweise automatisch, ggf. aber auch nur auf Anforderung. Werden Formeln eines Tabellenfeldes an andere Stellen kopiert, muss zwischen absolutem und relativem Zellbezug unterschieden werden. Formelzellen können auf andere Formelzellen verweisen. Mit diesem Prinzip können komplizierte Rechengänge mit vielen verknüpften Teil-Ergebnissen übersichtlich dargestellt werden.

Verschiedene Arten des Einsatzes von Tabellenkalkulationsprogrammen können sein:

- Automatisches Ausführen und Aktualisieren von Berechnungen
- Datenspeicherung & Datenverwaltung
- Automatisches erstellen und aktualisieren von Diagrammen
- Modellierung
- Programmierung (Makros, Visual Basic)

Einsatzmöglichkeiten

Vor allem aus dem kaufmännischen Bereich ist sie nicht mehr wegzudenken, wie ein Auszug aus den Anwendungsfeldern in der mittleren Spalte zeigt. Aber auch im wissenschaftlichen Bereich hat Tabellenkalkulation ihren Platz, insbesondere dann, wenn ein spezifischer Funktionsvorrat für Standardaufgaben aus den jeweiligen Bereichen angeboten wird (zB Statistikfunktionen, finanzmathematische Funktionen)

privat	betrieblich	wissenschaftlich
Haushaltsbudget	Kalkulationen	Statistik
Angebotsvergleich	einfache Kostenrechnung	Charts
Kostenvergleich	Reisekostenabrechnung	mathematische Modelle
Kreditkostenberechnung	Statistik	kaufmännische Anwendungen
uvm...	Charts	volkswirtschaftliche Modelle
	Investitionsrechnung	...uvm
	einfache Fakturierung	
	ABC-Analyse	
	Budgetierung	

Einschränkungen

Jedoch ist auch ein Tabellenkalkulationsprogramm keine „Eierlegende Wollmichsau“ und man stößt auch hier an Grenzen:

- Datenverwaltung wird schnell aufwendig ⇒ Lösung: Datenbanksystem verwenden
- Beschränkt in der Grösse ⇒ Lösung: Datenbanksystem verwenden
- Zusammenhänge sind nicht sichtbar ⇒ Lösung: Gute Dokumentation
- Gefahr von Nebenwirkungen (side effects) ist gross ⇒ Lösung: Sorgfältig arbeiten

Praxis

<https://support.office.com/de-de/excel>

Folgende Befehle/Funktionen sollte dir in MS Excel geläufig sein! Falls nicht erkunde die Hilfe von MS Excel.

- Öffnen/Speichern
- Navigieren mit Pfeiltasten & Tabulator
- Seite einrichten (Ränder, Ausrichtung, Format, Gitternetzlinien und Überschriften anzeigen/drucken)
- Arbeitsmappe
- Arbeitsblatt
- Zelle
- Namenfeld
- Spalte
- Zeile
- Bearbeitungsleiste/Formelzeile
- Automatisches Ausfüllen von Zellen
 - Sperren von Spalten und Zeilen
- Mehrere Spalten/Zeilen/Zellen markieren
- Breite/Höhe von mehreren Spalten/Zeilen ändern/automatisch an den Text anpassen
- Textausrichtung horizontal/vertikal
- Zellen verbinden
- Formatieren
- Bedingte Formatierung
- Sortieren und Filtern
- Grafiken einfügen
- Diagramme einfügen
- Daten einfügen
- Formeln
 - Grundrechnungsarten
 - Summe
 - Zählen
 - Anzahl
 - Anzahl2
 - Mittelwert
 - Maximum
 - Minimum
 - Median
 - Wenn
 - Oder
 - Und
 - Nicht
 - IstZahl
 - IstText
 - Summewenn
 - Zählenwenn
 - Verkettung
 - Links

- Rechts
- Teil
- SVerweis
- Mittelwertwenn
- ...
- Ansichten (Umbruchvorschau)
- Pivot-Tabellen und -Charts
- Fenster fixieren (Zeile, Spalte)
- Spalten ein- und ausblenden
- Duplikate entfernen
- Namen definieren
- Sekundärachse Diagramm
- Szenario Manager
- Datenüberprüfung
- Transponiertes Einfügen
- 3D-Summenfunktion
- Pivot Tabelle Gruppieren
- Erweitertes Filtern
- Teilergebnisse

Folgende Shortcuts sind sehr hilfreich:

Shortcut	Funktion
F2	Zelle editieren
STRG+Bild rauf/runter	Tabellenblatt wechseln
STRG+Z	Rückgängig
STRG+A	Alles markieren
STRG+C	Zelle kopieren
STRG+V	Zelle einfügen
STRG+X	Zelle ausschneiden
STRG+F	Suchen & Ersetzen
STRG+Mausklick	mehrere einzelne Zellen markieren
STRG+Pfeiltaste rechts	gehe zur ganz rechten Spalte
STRG+Pfeiltaste links	gehe zur ganz linken Spalte
STRG+Pfeiltaste unten	gehe zur ganz letzten Zeile
STRG+Pfeiltaste oben	gehe zur ganz ersten Zeile
SHIFT+Mausklick	mehrere Zellen neben- bzw. untereinander markieren
STRG+SHIFT+*	Ganze Tabelle markieren

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:6

Last update: **2020/05/12 10:12**



C++ bzw. CPP

- C++ Skriptum

- [Wiederholung 5. Klasse](#)
- [Funktionen](#)
- [Datentypen und -strukturen](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:7

Last update: **2020/05/26 08:57**



Funktionen in C++

Ein wichtiges Sprachelement von C++ kam bisher noch überhaupt nicht vor: die Funktion. Die Möglichkeit, Funktionen zu bilden, ist ein herausragendes Merkmal einer Programmiersprache. Ganz allgemein versteht man unter einer Funktion einen in sich geschlossenen Programmteil, der eine bestimmte Aufgabe erfüllt.

Der wichtigste Vorteil einer Funktion ist die WIEDERVERWENDBARKEIT von Programmcode.

Betrachten wir folgendes Beispiel:

```
int add( int x, int y)
{
    int z = x+y;

    return z;
}
```

An diesem Codeausschnitt erkennen Sie alle **Bestandteile einer Funktion**:

- **Typ des Rückgabewerts:** int
- **Rückgabewert:** z
- **Funktionsname:** add
- **Übergabeparameter:** (int x, int y)
- **Funktionskörper:** Anweisung innerhalb des Blocks der Funktion, begrenzt durch die geschweiften Klammern { und }
- **Return-Anweisung:** return z; (bei Rückgabetyyp void nicht nötig)

Zu all diesen Teilen ist natürlich noch Einiges zu sagen. Vorher noch ein typografischer Hinweis: Es hat sich eingebürgert, in Büchern zwei Klammern hinter Bezeichner zu setzen, die sich auf eine Funktion beziehen, zum Beispiel in Sätzen wie: Mit add() erreichen Sie die Addition zweier Ganzzahlen. Das sagt noch nichts über Art und Umfang der Argumentliste aus, sondern soll Sie lediglich daran erinnern, dass es dabei um eine Funktion und nicht um eine Variable geht. Ich will mich auch in diesem Buch daran halten.

Rückgabewert

In C++ muss jede Funktion einen Typ für den Wert angeben, den sie zurückliefert. Manchmal ist es aber auch gar nicht nötig oder sinnvoll, dass eine Funktion überhaupt einen Rückgabewert hat. In diesem Fall geben Sie als Typ void an.

Was macht man nun mit einem solchen Wert? Der Programmteil, der die Funktion aufruft, kann diese an allen Stellen einsetzen, wo er sonst eine Variable oder Konstante angeben würde (in obiger Form allerdings nur dort, wo lediglich der Wert benötigt wird), also etwa:

```
int main()
{
```

```
int a = 5;
int b = 12;
int c = add(a,b);

cout << "a = " << a << ", c = " << c << ", a+c = " << add(a,c) << endl;

return 0;
}
```

Dieses Programm hat dann die Ausgabe: a = 5, c = 17, a+c = 22

Übrigens: Selbst wenn eine Funktion einen Wert zurückgibt, müssen Sie ihn nicht beachten. Sie dürfen auch schreiben:

```
int a = 5;
int b = 12;
add(a,b);
```

auch wenn das hier keinen Sinn machen würde. Bei Funktionen mit Rückgabetypp void ist das hingegen die übliche Form des Aufrufs. Allgemein kommt es aber häufiger vor, dass Rückgabewerte ignoriert werden. Beispielsweise geben viele Funktionen Statusinformationen darüber zurück, wie gut (oder schlecht) sie ihre Aufgabe erfüllen konnten. Viele Anwender solcher Funktionen interessieren sich nicht für den Status und übergehen ihn. Das kann manchmal aber auch gefährlich werden, wenn etwa aufgetretene Fehler aus diesem Grund zunächst unentdeckt bleiben.

Funktionsname

Wie alle anderen Bezeichner in C++ dürfen Sie auch Funktionsnamen nur aus Buchstaben, Ziffern sowie dem Unterstrich _ bilden. Außerdem ist es nicht erlaubt, Funktionsnamen zu verwenden, die Schlüsselwörtern gleichen (etwa for).

Übergabeparameter

Eine Funktion kann immer nur auf den Daten arbeiten, die ihr lokal vorliegen. Außer global (das heißt außerhalb aller Funktionen) definierten Variablen sind das nur die Parameter, die das Hauptprogramm an die Funktion übergibt. Von diesen Parametern (auch Argumente genannt) können Sie keinen, einen oder mehrere angeben, die Sie dann durch Kommas trennen.

Wenn Sie eine Funktion ohne Argumente schreiben wollen, lassen Sie den Bereich zwischen den beiden runden Klammern einfach leer – denn die Klammern müssen Sie stets schreiben! – oder Sie setzen ein Argument vom Typ void ein.

Ansonsten geben Sie für jeden Parameter seinen Datentyp und einen Namen an, unter dem er in der Funktion bekannt sein soll. Dieser Name kann vollkommen anders sein, als der im Hauptprogramm beim Aufruf verwendete. Auch in obigem Beispiel heißen die Summanden in der Funktion x und y, im Hauptprogramm aber a und b.

Funktionskörper

Hier stehen die Anweisungen, die bei einem Aufruf der Funktion ausgeführt werden. Man kann darüber streiten, wie lang Funktionen sein sollten. Es gibt Experten, die fordern, dass eine Funktion aus nicht mehr als 50 Zeilen bestehen dürfe, sonst werde sie unleserlich. Es gibt jedoch in der Praxis immer wieder Fälle, in denen längere Funktionen sinnvoll sind. Bei der objektorientierten Programmierung werden Sie allerdings ohnehin wesentlich mehr Funktionen (beziehungsweise Methoden) verwenden, die im Durchschnitt wesentlich kürzer sind als bei der prozeduralen Programmierung.

Innerhalb des Funktionskörpers können Sie die Funktionsparameter wie normale Variablen verwenden; zusätzlich können Sie natürlich auch noch lokale Variablen definieren. Außerdem ist es selbstverständlich erlaubt, aus einer Funktion wieder andere Funktionen aufzurufen. (Sie dürfen sogar die Funktion selbst wieder aufrufen; man spricht dann von einer rekursiven Funktion – aber das ist ein eigenes Thema.)

Return-Anweisung

Die Anweisungen im Funktionskörper werden so lange abgearbeitet, bis das Programm auf das Ende der Funktion oder eine return-Anweisung trifft. Diese erfüllt einen doppelten Zweck:

Sie legen fest, welchen Wert die Funktion an das Hauptprogramm zurückliefern soll. Das kann eine Variable sein oder ein Ausdruck, eine Konstante oder der Rückgabewert einer anderen Funktion (wobei Letzteres als schlechter Stil gilt). Hat Ihre Funktion den Rückgabebetyp void, geben Sie an dieser Stelle überhaupt nichts an. Der Typ des angegebenen Werts muss jedoch in jedem Fall mit dem deklarierten Rückgabebetyp übereinstimmen. Sie beenden die Funktion und kehren zum Hauptprogramm zurück. Jede return-Anweisung – sei sie nun am Ende oder irgendwo inmitten des Funktionskörpers – markiert das Ende der Abarbeitung der Funktion und den Rücksprung an die Stelle, von der aus die Funktion aufgerufen wurde. Sie können also die Funktion schon beenden, bevor alle Anweisungen ausgeführt sind, zum Beispiel wenn eine bestimmte Bedingung erfüllt ist. Ist der Rückgabebetyp void, muss am Ende der Funktion keine return-Anweisung stehen (auch nicht das Schlüsselwort return), wie in folgendem Beispiel:

```
void ausgabe( int z )
{
    cout << ``Das Ergebnis ist: `` << z << endl;
}
```

Wenn Sie allerdings bei Funktionen mit irgendeinem anderen Rückgabebetyp die return-Anweisung vergessen, meldet der Compiler einen Fehler.

Funktionsdeklaration bzw. Funktionsprototyp

Bevor Sie eine Funktion verwenden können, müssen Sie dem Compiler zunächst mitteilen, dass es eine Funktion dieses Namens gibt, wie viele und welche Parameter sie hat und welchen Typ sie zurückliefert. Dies geschieht mit einem so genannten Prototyp der Funktion. Der Prototyp sieht

genauso aus wie die Funktion selbst bis auf den Funktionskörper; dieser fehlt und wird durch ein einfaches Semikolon ; ersetzt. (Es ist sogar erlaubt, die Namen der Argumente wegzulassen und nur ihre Typen anzugeben. Analog zu den Klassen (siehe Seite [*]) ist also der Prototyp die Deklaration und die Funktion mit Körper die Definition.

Funktionsaufruf

Der Aufruf einer Funktion erfolgt durch Nennung des Funktionsnamens. An den Funktionsnamen schließt sich immer ein Klammerpaar an, das gegebenenfalls auch Parameter enthalten kann. Dieses Klammerpaar ist zwingend erforderlich. Die Parameter des Aufrufs müssen zu den Parametern der Funktion passen. Besitzt die Funktion einen Rückgabewert, kann der Funktionsaufruf als Ausdruck verwendet werden. Er kann also beispielsweise auf der rechten Seite einer Zuweisung stehen.

Übergabe der Argumente

C++ kennt mehrere Varianten, wie einer Funktion die Argumente übergeben werden können: call-by-value, call-by-reference und call-by-pointer.

call-by-value (Wertübergabe)

Bei call-by-value (Wertübergabe) wird der Wert des Arguments in einen Speicherbereich kopiert, auf den die Funktion mittels Parametername zugreifen kann. Ein Werteparameter verhält sich wie eine lokale Variable, die „automatisch“ mit dem richtigen Wert initialisiert wird. Der Kopiervorgang kann bei Klassen (Thema eines späteren Kapitels) einen erheblichen Zeit- und Speicheraufwand bedeuten!

Beispiel

```
#include <iostream>
#include <conio.h>

using namespace std;

//Prototyp, Deklaration
int sumW(int x,int y);    //Parameterübergabe per Wert (Value)

int main(int argc, char** argv) {
    cout << "Hello World" << endl;

    int a=5,b=6,erg=0;
    cout << "Parameteruebergabe per Wert" << endl;
    erg=sumW(a, b);
    cout << "a: " << a <<endl;
    cout << "b: " << b <<endl;
```

```

    getch();

    return 0;
}

//Funktionendefinition - Parameterübergabe per Wert
int sumW(int x,int y){
    x+=1; //x=x+1; x++; ++x;
    y+=1;

    cout << "x: " << x <<endl;
    cout << "y: " << y <<endl;

    return x+y;
}

```

Beispielhafte Speicherbelegung des Programms im Hauptspeicher:

Variable	Adresse	Wert
a	0x0001	5
b	0x0005	6
...	.	.
...	.	.
x	0x00a1	5 → 6
y	0x00a5	6 → 7

Die Ausgabe des Programms ist:

```

Hello World
Parameteruebergabe per Wert
x: 6
y: 7
a: 5
b: 6

```

call-by-reference (Übergabe per Referenz)

Sollen die von einer Funktion vorgenommen Änderungen auch für das Hauptprogramm sichtbar sein, müssen in C sogenannte Zeiger verwendet werden. C++ stellt ebenfalls Zeiger zur Verfügung. C++ gibt Ihnen aber auch die Möglichkeit, diese Zeiger mittels Referenzen zu umgehen, was im alten C nicht möglich war. Beide sind jedoch noch Thema eines späteren Kapitels.

Im Gegensatz zu call-by-value wird bei call-by-reference die Speicheradresse des Arguments übergeben, also der Wert nicht kopiert. Änderungen der (Referenz-)Variable betreffen zwangsläufig auch die übergebene Variable selbst und bleiben nach dem Funktionsaufruf erhalten. Um call-by-reference anzuzeigen, wird der Operator & verwendet, wie Sie gleich im Beispiel sehen werden. Wird

keine Änderung des Inhalts gewünscht, sollten Sie den Referenzparameter als const deklarieren, um so den Speicherbereich vor Änderungen zu schützen. Fehler, die sich aus der ungewollten Änderung des Inhaltes einer übergebenen Referenz ergeben, sind in der Regel schwer zu finden.

```
#include <iostream>
#include <conio.h>

using namespace std;

//Prototyp, Deklaration
int sumR(int &x,int &y);    //Parameterübergabe per Referenz (Reference)

int main(int argc, char** argv) {
    cout << "Hello World" << endl;

    cout << "Parameteruebergabe per Referenz" << endl;
    a=5,b=6,erg=0;
    erg=sumR(a, b);
    cout << "a: " << a <<endl;
    cout << "b: " << b <<endl;

    getch();

    return 0;
}

//Funktionendefinition - Parameterübergabe per Referenz
int sumR(int &x,int &y){
    x+=1; //x=x+1; x++; ++x;
    y+=1;

    cout << "x: " << x <<endl;
    cout << "y: " << y <<endl;

    return x+y;
}
```

Beispielhafte Speicherbelegung des Programms im Hauptspeicher:

Variable	Adresse	Wert
x,a	0x0001	5 → 6
y,b	0x0005	6 → 7
...
...
...
...

Die Ausgabe des Programms ist:

```
Hello World
Parameteruebergabe per Referenz
x: 6
y: 7
a: 6
b: 7
```

call-by-pointer (Übergabe per Zeiger)

Wenn Sie einen Zeiger als Parameter an eine Funktion übergeben, können Sie den Wert an der übergebenen Adresse ändern.

```
#include <iostream>
#include <conio.h>

using namespace std;

//Prototyp, Deklaration
int sumZ(int *x,int *y); //Parameterübergabe per Zeiger (Pointer)

int main(int argc, char** argv) {
    cout << "Hello World" << endl;
    cout << "Parameteruebergabe per Zeiger" << endl;
    a=5,b=6,erg=0;
    erg=sumZ(&a, &b);
    cout << "a: " << a <<endl;
    cout << "b: " << b <<endl;

    getch();

    return 0;
}

//Funktionendefinition - Parameterübergabe per Zeiger
int sumZ(int *x,int *y){
    *x+=1; //x=x+1; x++; ++x;
    *y+=1;

    cout << "x: " << *x <<endl;
    cout << "y: " << *y <<endl;

    return *x+*y;
}
```

Beispielhafte Speicherbelegung des Programms im Hauptspeicher:

Variable	Adresse	Wert
a	0x0001	5 → 6
b	0x0005	6 → 7
...
...
*x	0x00a1	0x0001
*y	0x00a5	0x0005

Die Ausgabe des Programms ist:

```
Hello World
Parameteruebergabe per Zeiger
x: 6
y: 7
a: 6
b: 7
```

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:7:7_01



Last update: **2020/05/27 07:48**

Datentypen und -strukturen

Um **Daten der Realität** abbilden zu können, **reichen die Basistypen der Sprache C++ nicht aus**. Es ist aber möglich, diese Typen zusammenzustellen. Durch das Zusammensetzen der Basistypen zu neuen Datenstrukturen versucht der Programmierer, die **Daten der Wirklichkeit in seinem Programm nachzubilden**.

- Aufzählungstyp enum
- Das Array
- Der Zeiger und die Adresse
- Der Variablenverbund: struct
- Dynamische Strukturen

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:7:7_02

Last update: **2020/05/26 08:57**

