

 [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/Okttober)

## Projekt

- Projekt zu Videoschnitt (Oktober/November)

## Kapitel

- 1) Audiobearbeitung
- 2) Videobearbeitung
- 3) Web-Techniken (HTML & CSS)
- 4) Textverarbeitung
- 5) Tabellenkalkulation
- 6) C++ Arrays, Sortieralgorithmen, Strukturen & Klassen
- 7) C# Visuelles Programmieren
- 8) 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 - Mai**

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

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

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

Kapitel 1 bis inkl. 3.4.4

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



Permanent link:  
[http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi\\_201920](http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920)

Last update: **2019/11/17 16:03**

# Was wird in der 6. Klasse gemacht?

## 3. Semester

### Sicherung der Nachhaltigkeit

- Notwendiges Vorwissen 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-Zugänge 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 graphischen 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 Vorwissen 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](http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:0_lehrplaninhalte)



Last update: **2019/09/06 20:55**

# 1) Audiobearbeitung

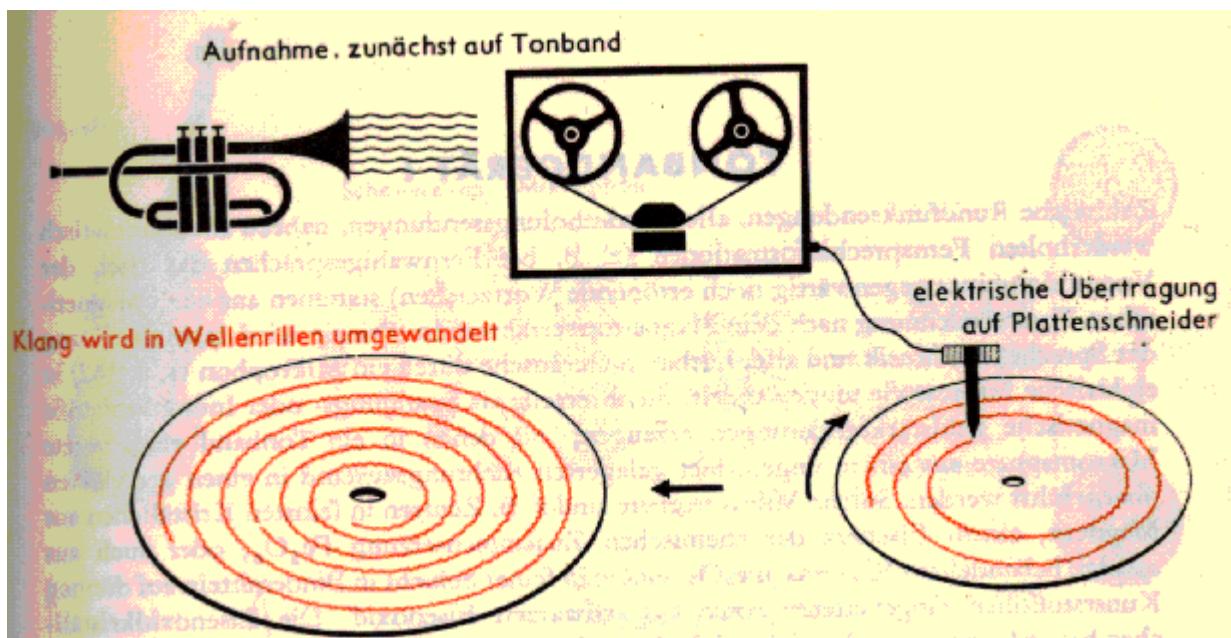
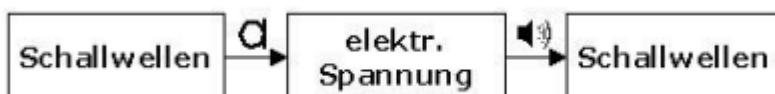
## 1.1) Theorie

### 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 Audiobereich 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?!).

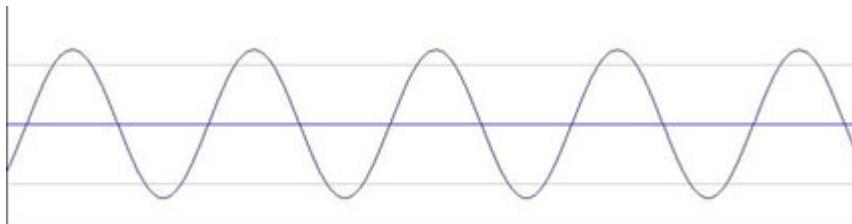
## Sprache

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.

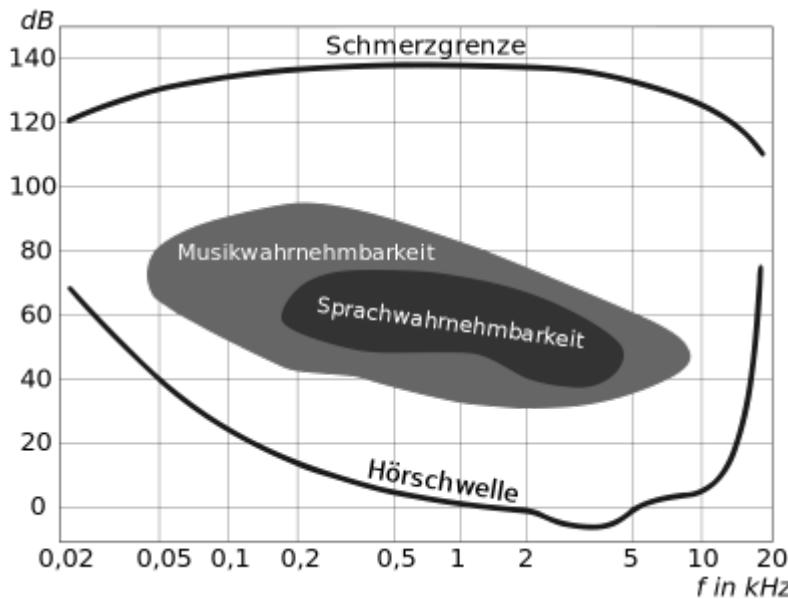
## Beispiele für Schwingungen im Audibereich

### Sinuswelle

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



### Hörbereich



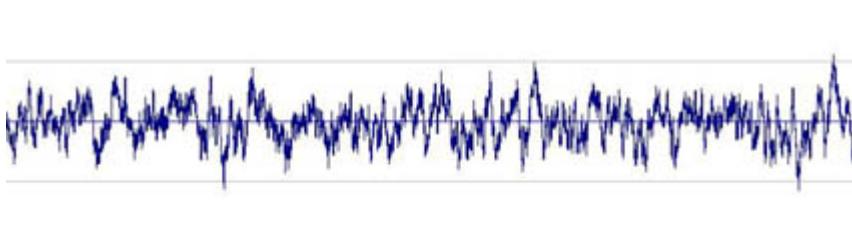
Sinuswellenton in verschiedenen Frequenzen → Teste dein Gehör!



## Sinuswelle in verschiedenen Frequenzen (20Hz bis 20kHz)

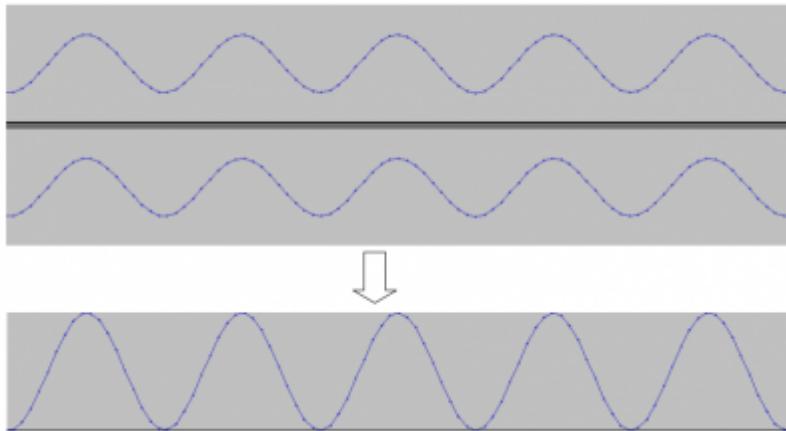
### Rauschen

Schwingungschaos = das Gegenteil einer periodischen Schwingung.

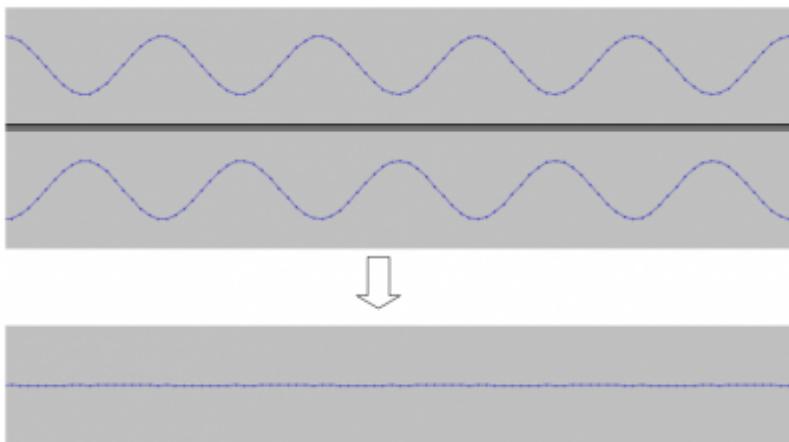


### Kombination von Tonquellen

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



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



## Maskierungseffekt



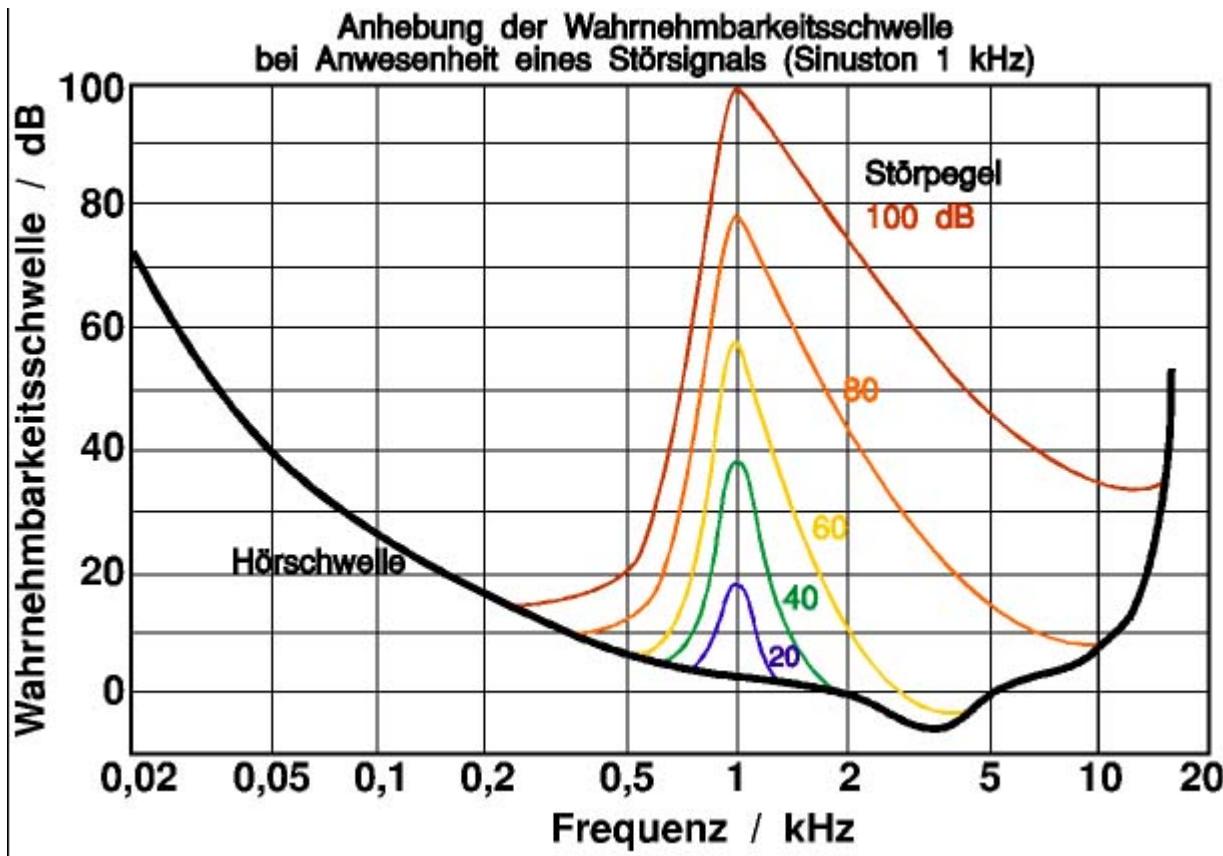
# Maskierungseffe kt

Maskierungseffekte (auch Verdeckung genannt) bewirken beim menschlichen Gehör, dass der Mensch in einem Geräusch bestimmte Frequenzanteile nicht oder nur mit verringelter 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 erahnen lässt, Audiosignale oberhalb einer bestimmten (möglicherweise einstellbaren) Frequenz durch, während Frequenzen darunter abgedämpft werden.

## Tiefpassfilter

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

# Digitalisierung von Audio

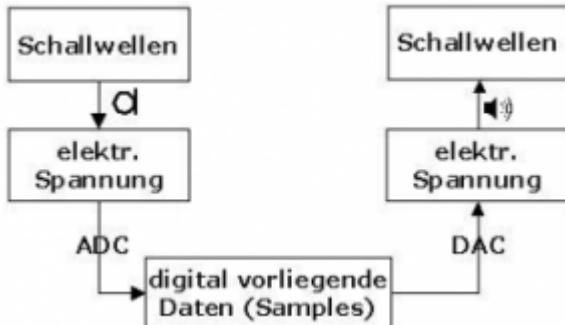
Grundsätzlich sind Daten im Audiobereich zeitkritisch, d.h. sie stellen Anforderungen an Hard- und Software, die historisch gesehen atypisch für einen 'Rechner' sind. Bis heute sind damit Geschwindigkeits-, Bandbreiten- und Speicherprobleme verbunden, die von technischen und ökonomischen Rahmenbedingungen 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.R Regel von der Soundkarte zur Verfügung gestellt.

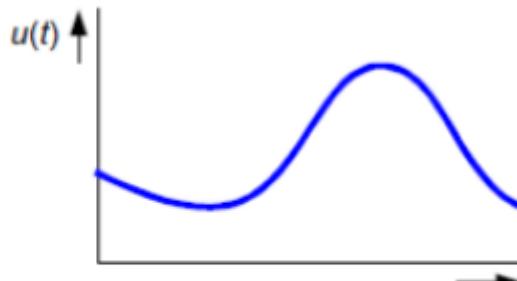
- hierbei wird die analoge Spannung in einem regelmäßigen Zeittakt (**Samplingfrequenz**) gemessen.
- Diese Messwerte sind je nach ausgewählter **Bitrate** aufgelöst
- heraus kommt eine zahlencodierte Nachbildung des analogen Signals als eine **Abfolge von diskreten Schritten**

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



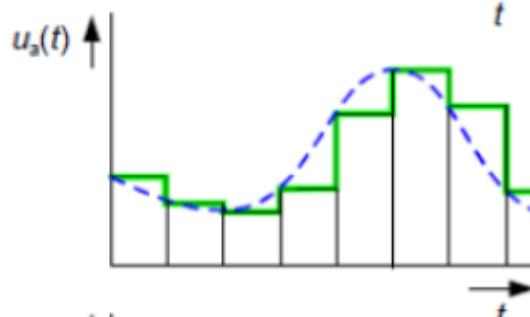
Beim Computer befinden sich die beiden Converter in der Soundkarte oder sind schon direkt im Mainboard integriert.

**Kontinuierliches Signal:**



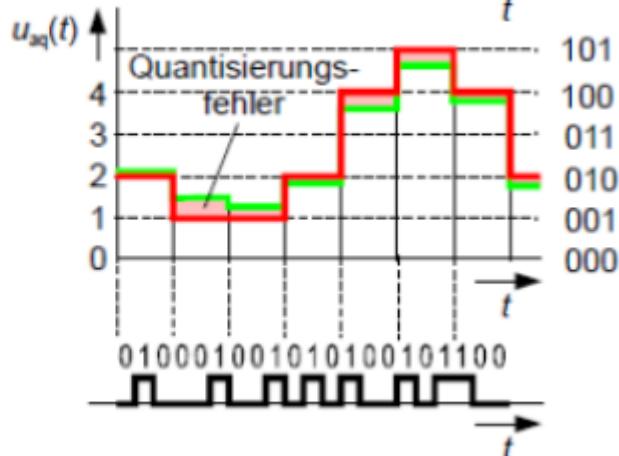
**Abgetastetes Signal:**

(S&H 100%)



**Quantisiertes Signal:**

(lineare Quantizierung)



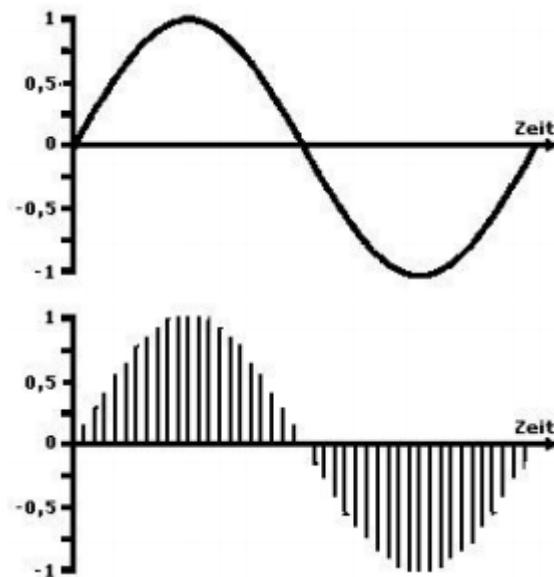
**Digitalisiertes Signal:**

## Kanäle

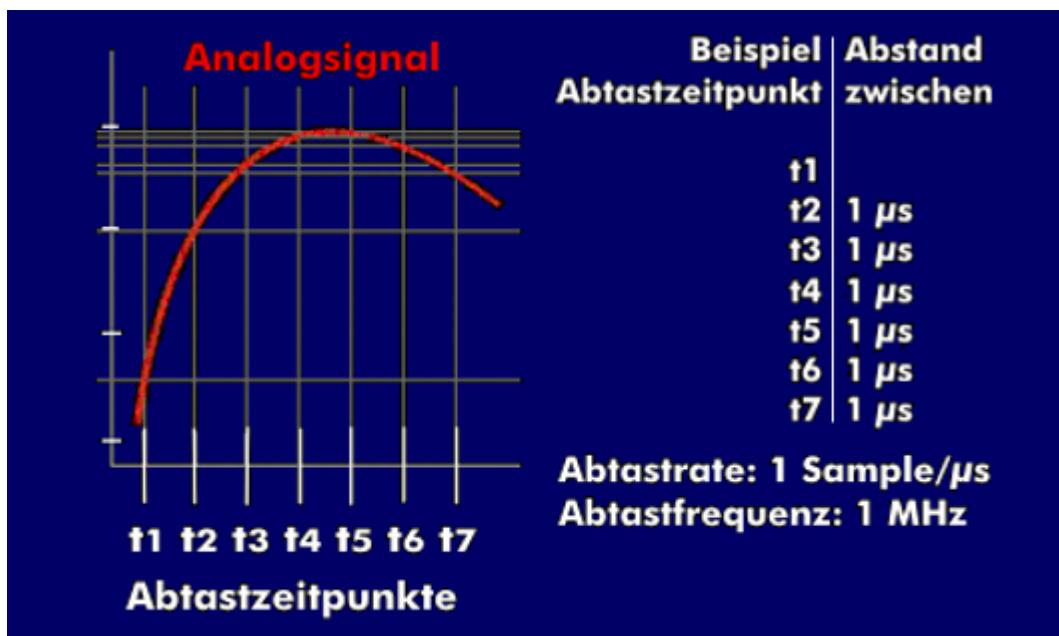
Eine Audiodatei kann mehrere Kanäle haben. Verbreitet sind Mono (ein Kanal), Stereo (2 Kanäle) und 5.1 bzw. 7.1 (Surround). Ein Kanal liefert jeweils die Informationen für einen der Lautsprecher und ist ein eigenständiges Audiosignal. Das heißt wir können eine Stereo-Datei aufteilen und in zwei Mono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

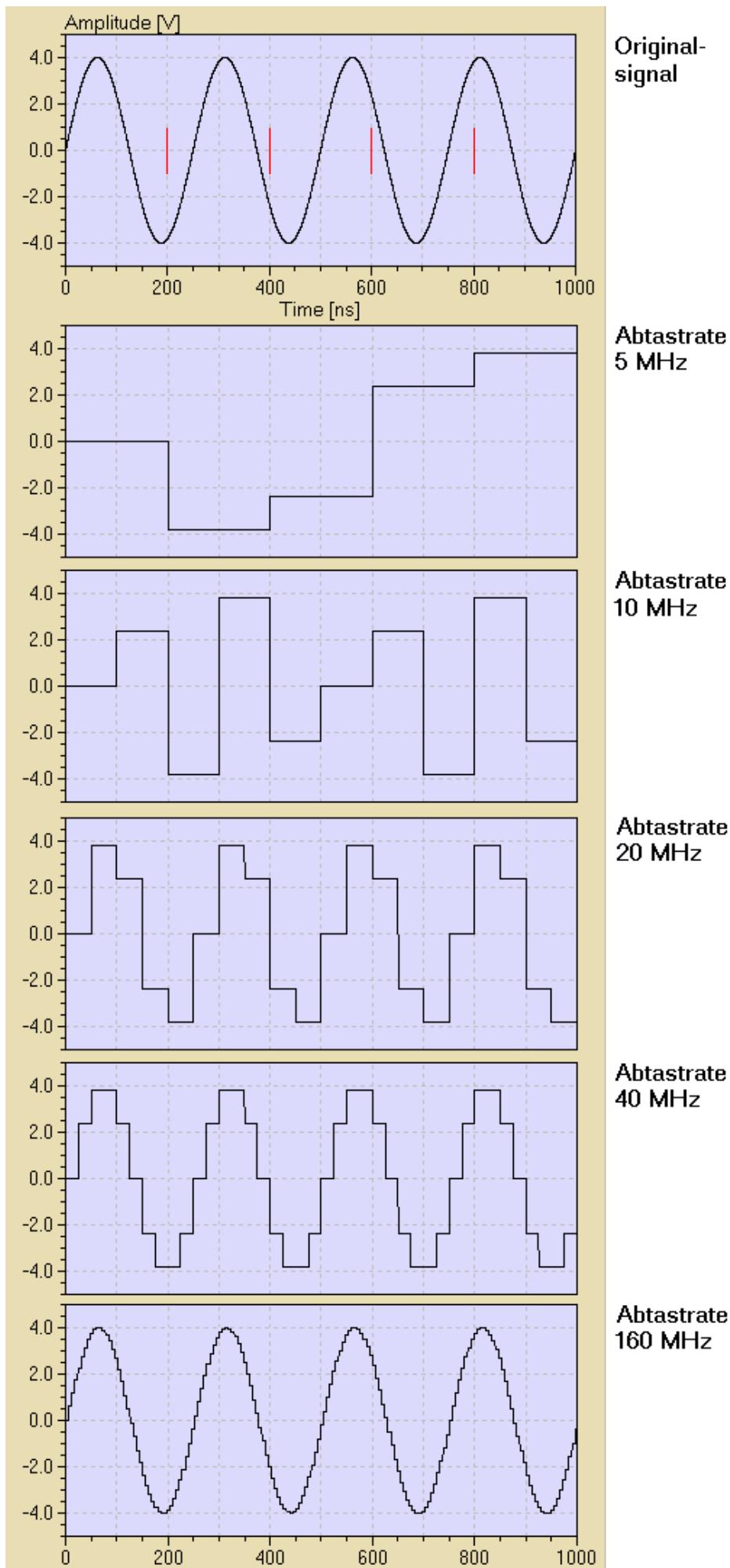
### Beispiel

Das folgende Originalsignal zeigt in der Skizze eine Sinuswelle mit der Schwingungsdauer von T=250ns.

Das heißt die Frequenz (`1/s = Hz`) des Originalsignals ist:

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

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.



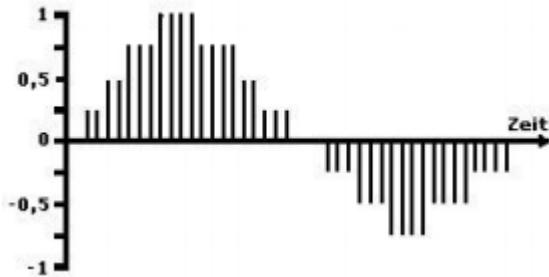
## Bittiefe

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 ->  $2^1 = 2$  Zustände (0 oder 1)
- 2 Bit ->  $2^2 = 4$  Zustände (0 bis 3)
- 3 Bit ->  $2^3 = 8$  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:

<b>Qualität</b>	<b>Bittiefe</b>	<b>Samplingfrequenz Hz</b>	<b>Anzahl Kanäle</b>	<b>Downloadrate kbit/s</b>	<b>Speicherbedarf MByte/min</b>
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:

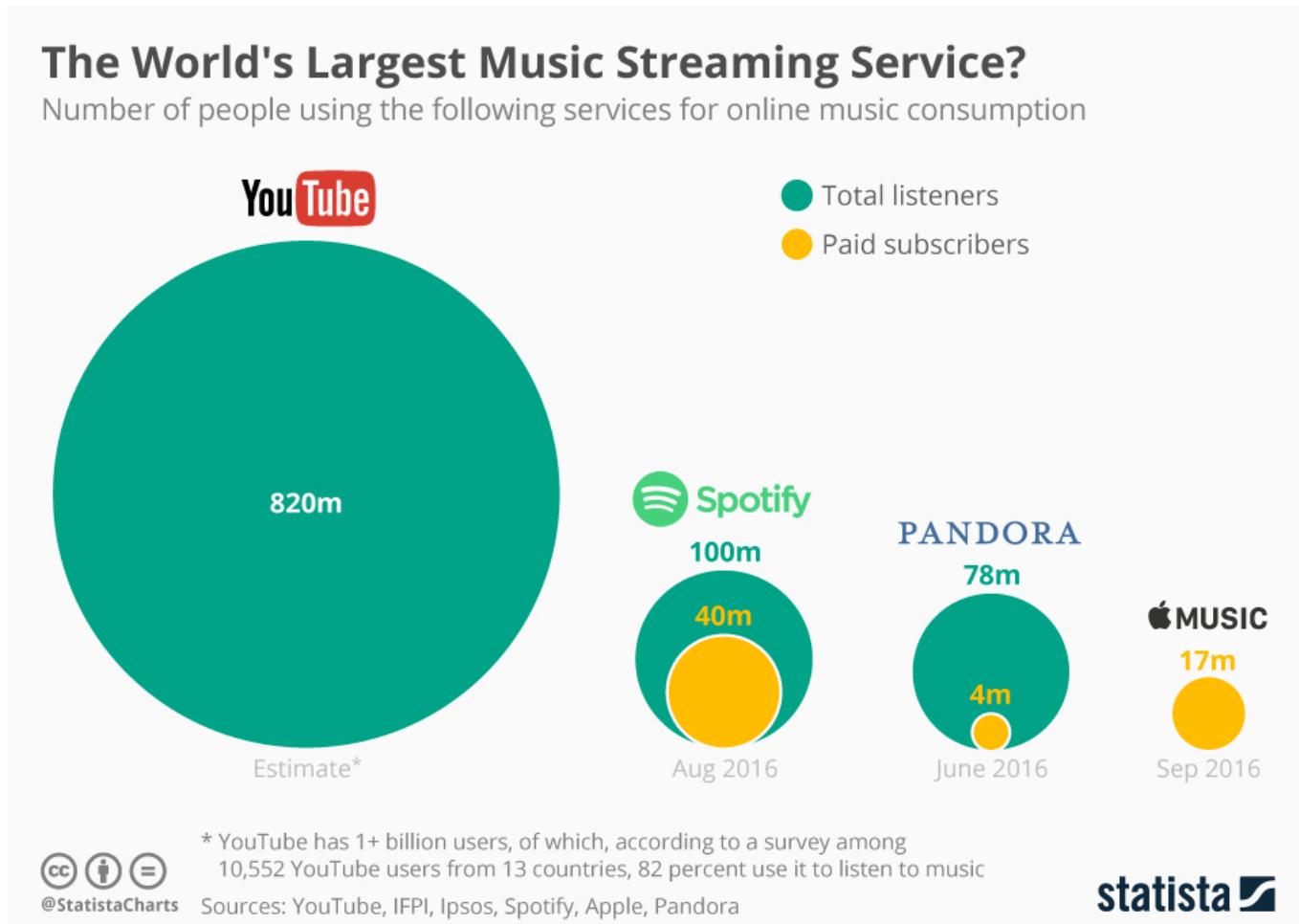
<b>Anwendung</b>	<b>Qualität</b>	<b>Codecs</b>	<b>Original kbit/s</b>	<b>Komprimiert kbit/s</b>	<b>Kompressionsfaktor</b>
Internet Echtzeit-Streaming	Telefon bis Radio	mp3, mpeg4, wma, RealAudio, ogg, vorbis	689-1378	30-60	ca. 11-45
Internet Download	quasi-CD	mp3, wma, ogg, vorbis	1378	128	10,7
DVD	quasi-DVD	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.

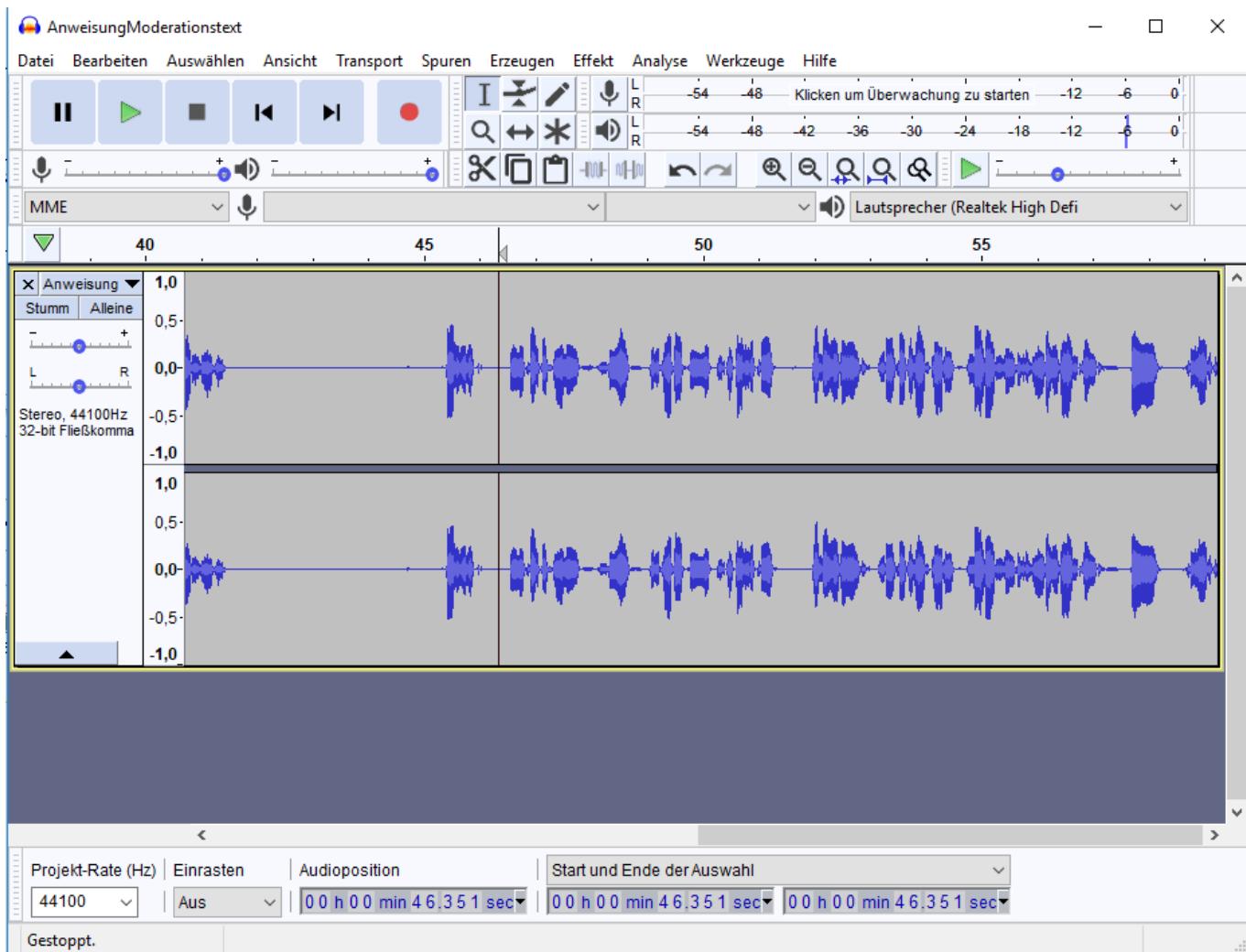


## 8.2) Praxis mit Audacity

Die Audiobearbeitung 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 Audiobearbeitung, 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-destructive Audiotools

Sehr wichtig für deine Audiodateien ist die Unterscheidung zwischen **destruktiver** und **non-destructiver Audiotools**. 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-destructiven Audiotools** 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 (bounce)** – in eine neue Datei. Dein **Original bleibt jederzeit unangetastet** und Du kannst schnell zum Ursprung zurückkehren.

### Vorteile non-destructiv

- 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 Audiobearbeitung 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.

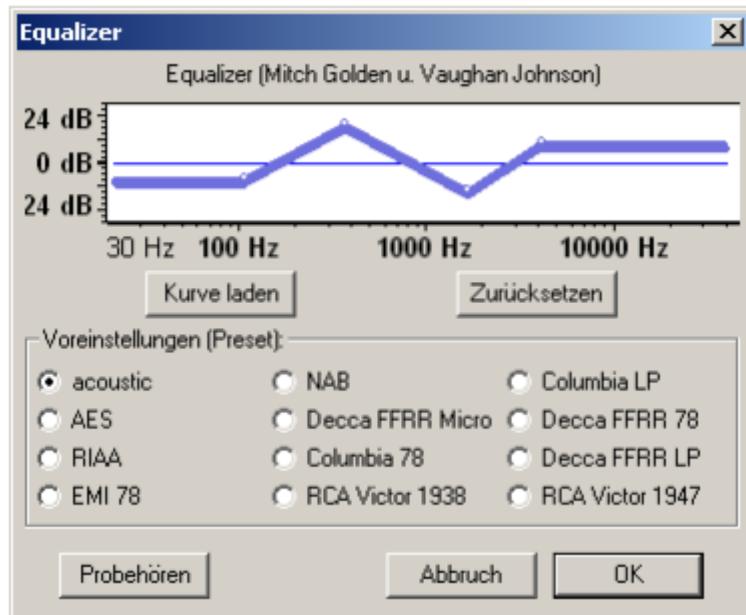
## Normalisieren von Audio

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 gleichmäßige 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 rausgefahrt 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 Knacken 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 Audiotools 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

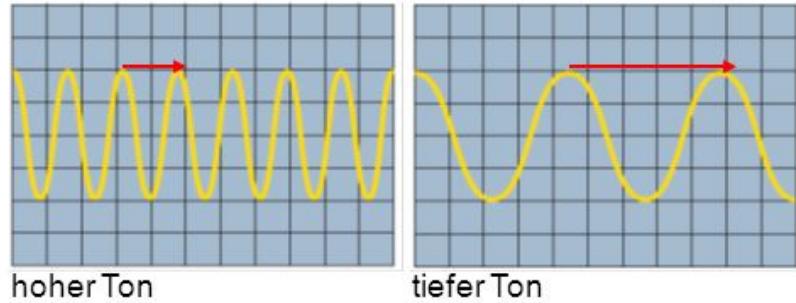
Das Tempo, auch Zeitmaß, gibt in der Musik an, wie schnell ein Stück zu spielen ist, bestimmt also die absolute Dauer der Notenwerte.

## 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.

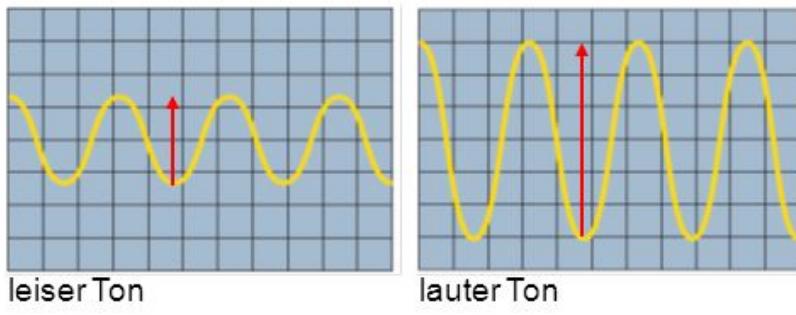


## Lautstärke ändern

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.



## **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](http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:1)

Last update: **2019/09/16 12:27**

# 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](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 Audiobearbeitung
- 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](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

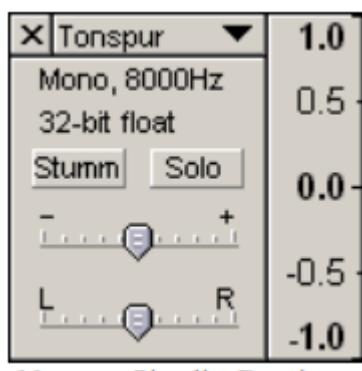


Permanent link:  
[http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi\\_201920:1:uebungen:3](http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:1:uebungen:3)

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

## Übung 4: Audioaufnahme

- a) Erstellen Sie ein neues Projekt in Audacity. Schließen Sie ein Mikrophon 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



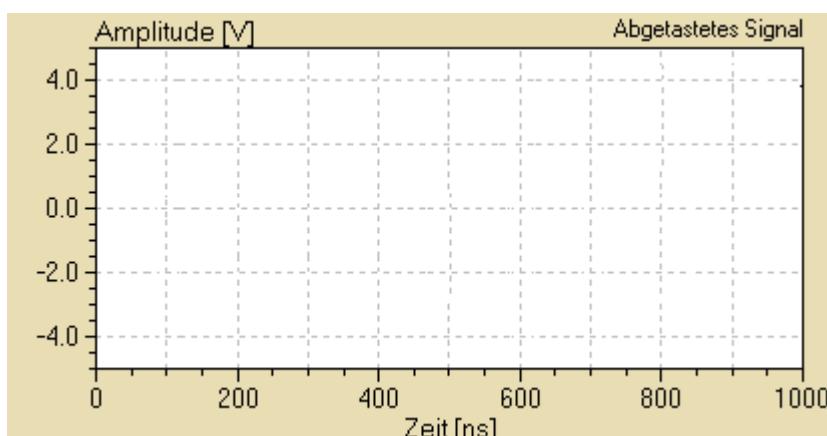
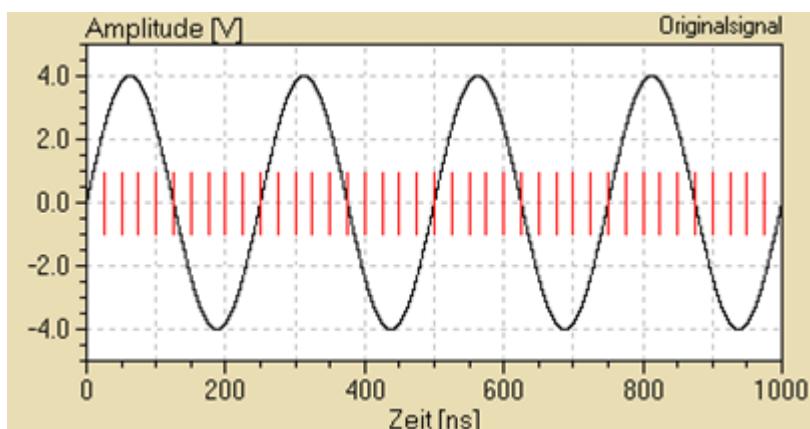
Permanent link:  
[http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi\\_201920:1:uebungen:4](http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:1:uebungen:4)

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](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.

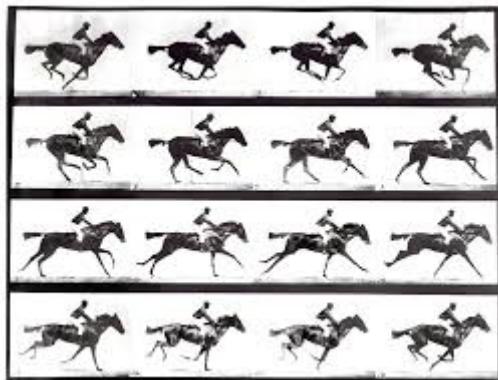
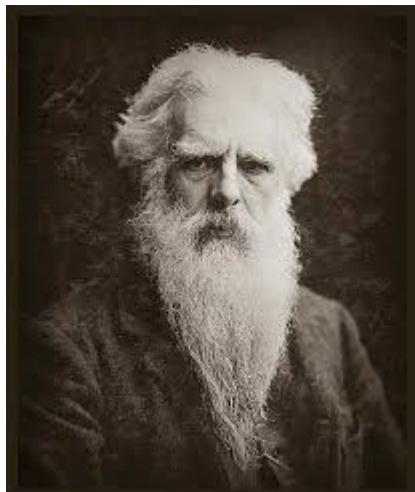
Es gelten **ähnliche fototechnische Prinzipien** (Blende, Belichtungszeit, Dynamik...) wie bei einer Spiegelreflex(DSLR/M)-Kamera. Zusätzlich wird jedoch der Datenstrom in ein bestimmtes Format **kodiert bzw. dekodiert (Codec)**, zB. MPEG oder h.264 und gespeichert. Als **Speichermedien** dienen Film, 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 **Nachbildungswirkung** (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. Jhd.)** 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**:  $1920 \text{ px} \times 1080 \text{ px} = 2\,073\,600 \text{ px}$
- **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.

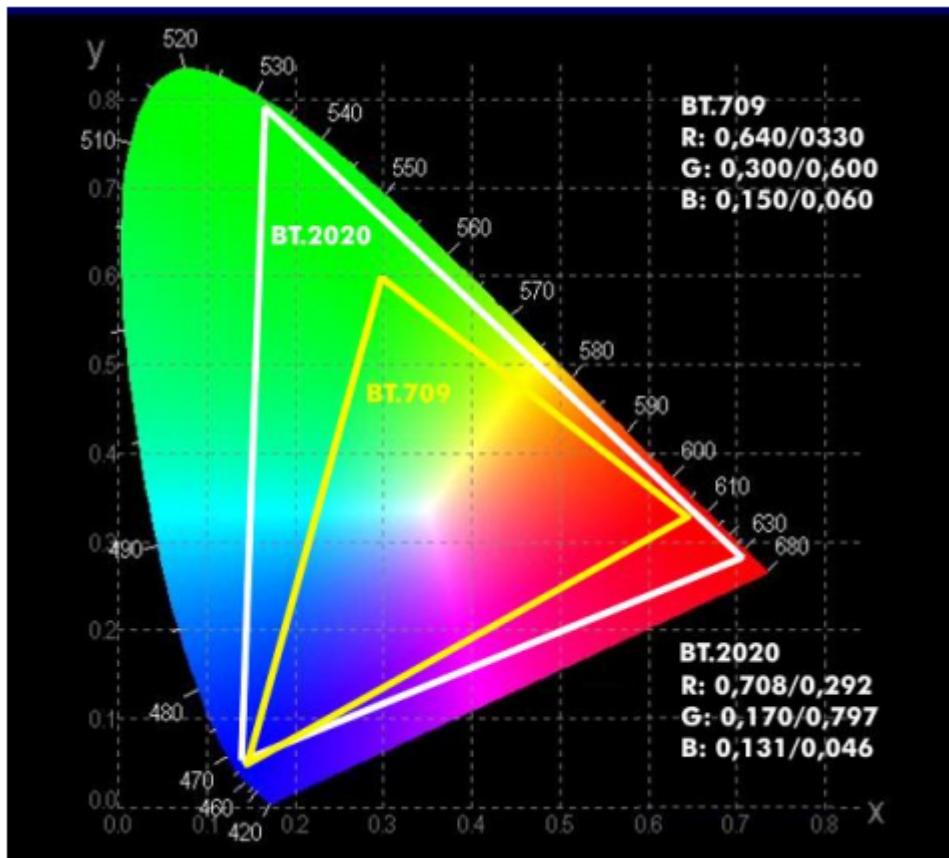
### HDTV-Format

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.

	<b>BT.709</b>	<b>BT.2020</b>
<b>Bildpunkte</b>	HD	UHD, 8K
<b>Bildwiederholfrequenz</b>	25, 24, 30, 50, 60	24, 25, 30, 50, 60, 100, 120
<b>Zeilensprung/Progressiv</b>	Beide	Nur progressiv
<b>Farbraum</b>	BT.709	BT.2020
<b>Bittiefen</b>	8, 10	10, 12
<b>Farbmodell</b>	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 Bildwechselraten 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 Bildwechselraten 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 Bildwechselraten (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				
1080i/30	1080	1125	1920	2200
1080p/30				
720p/50	720	750	1280	1980

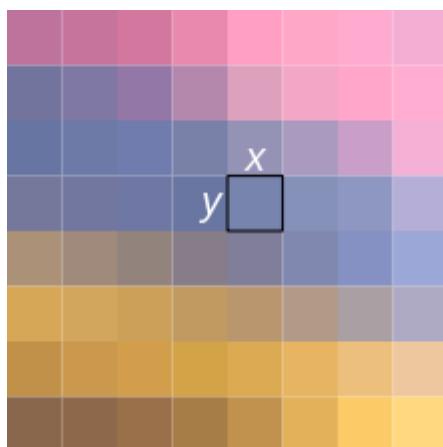
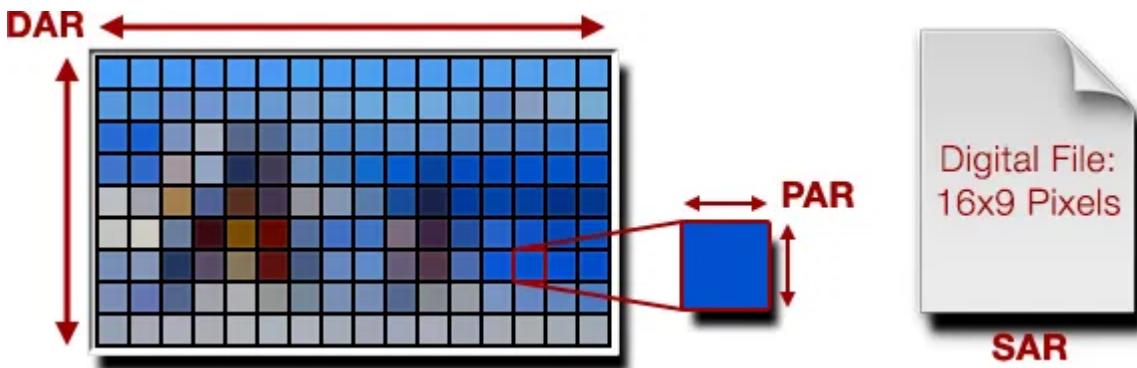
HD-Format		Auflösung sichtbar		Austastung		Gesamt	
	Bildfrequenz (Hz)	Horizontal (Pixel)	Vertikal (Pixel)	H-Blank (Pixel)	V-Blank (Pixel)	Horizontal (Pixel)	Vertikal (Pixel)
1080p/24	24	1920	1080	830	45	2750	1125
1080i/25	25 (2 Halbb.)	1920	1080	720	45	2640	1125
1080p/25	25	1920	1080	720	45	2640	1125
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

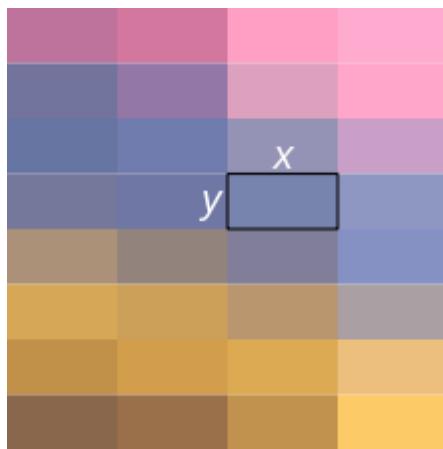
Berechnung der Bruttodatenraten anhand 1080i bei 10 bit:

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

## Seitenverhältnis

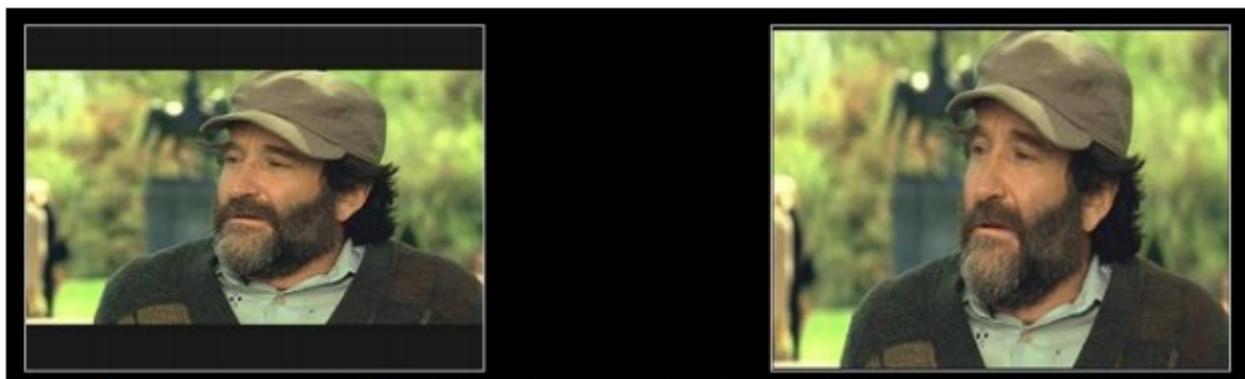
**Aspect Ratio** = Seitenverhältnis. Unterscheide SAR (Storage Aspect Ratio), DAR (Display Aspect Ratio) und PAR (Pixel Aspect Ratio). Wenn die Seitenverhältnisse für SAR und DAR übereinstimmen spricht man von **quadratischen Pixeln (square pixel)**, falls nicht, von **nicht-quadratischen Pixeln (nonsquare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

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

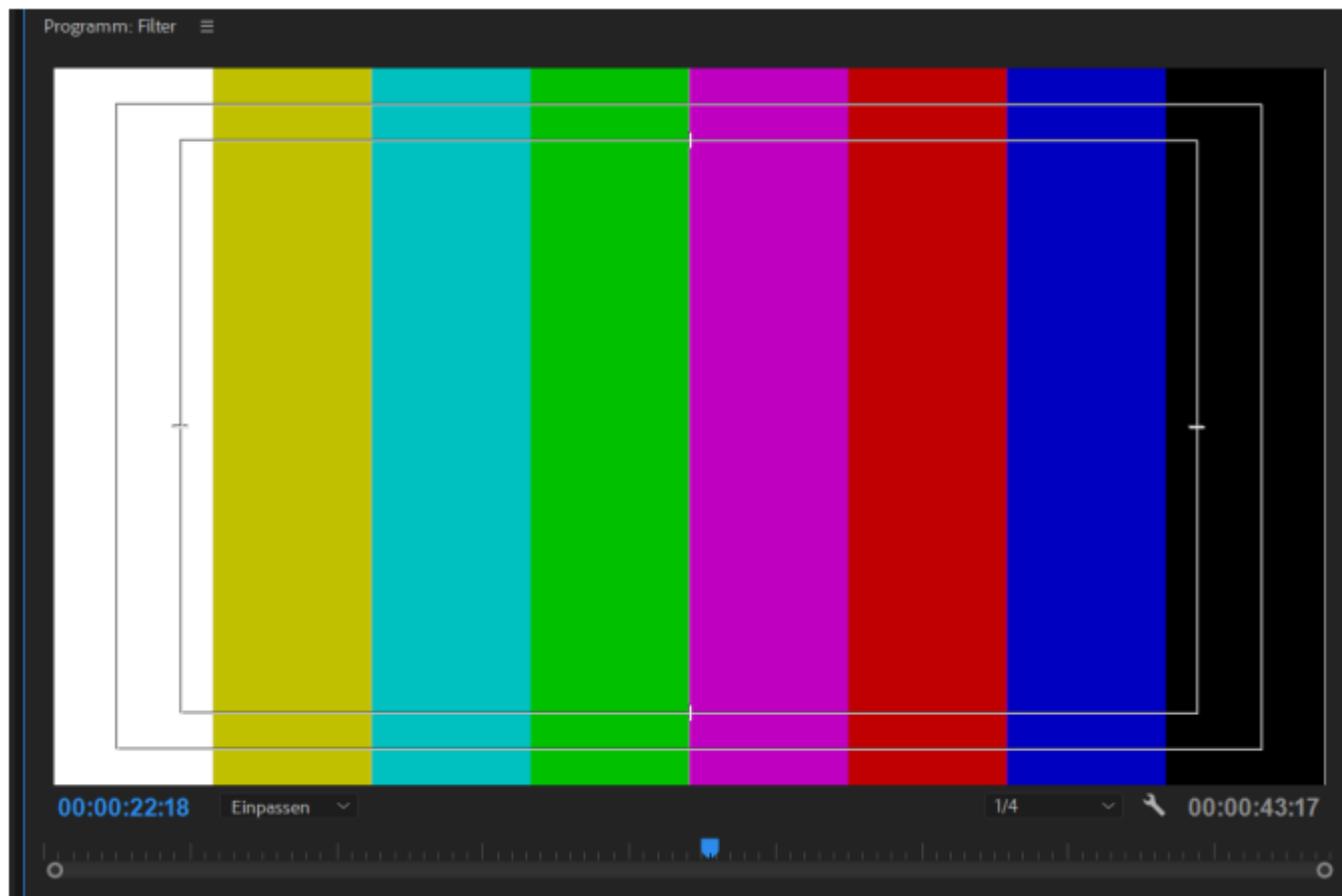


Nicht-Anamorph und Anamorph. Links ist die entzerrte Version.

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.



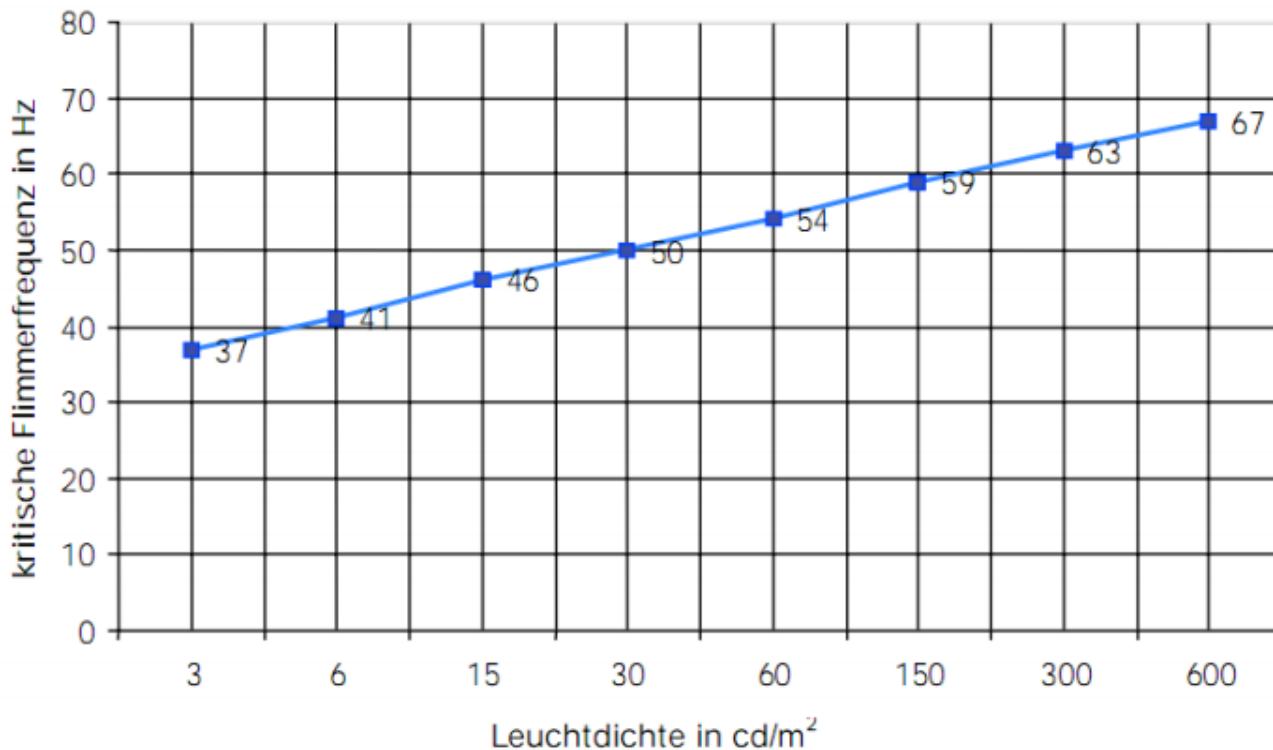
## Bildverschmelzungsgrenze und Bildwiederholungsfrequenz

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 **Bildwiederholrate** 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 Verschub und eine **höhere zeitliche Auflösung** (=flüssiger Bewegungseindruck bei gleicher Bandbreite). Dafür ist das einzelne Bild nicht so scharf und kann bei bewegungsreichen Aufnahmen Zeilenflimmern auftreten.



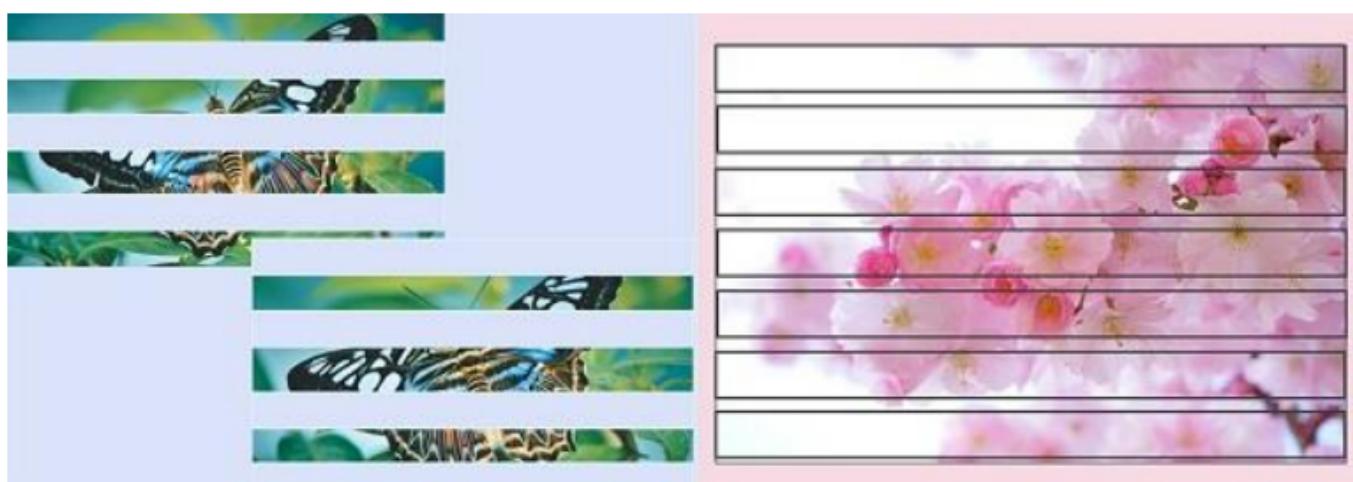
## Video zu Interlaced, Progressive & Bildwiederholrat en]



## 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 Verschub 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



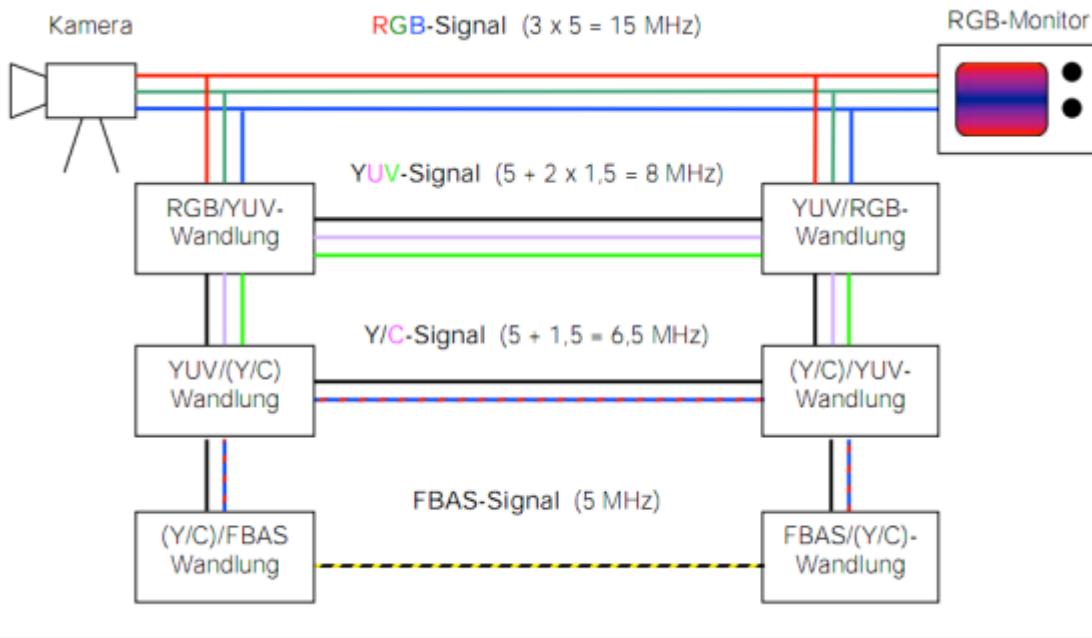
Links: Zwei Halbbilder vereint (50i), Rechts: Progressive Abtastung (25p)

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 Signalfluss 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 Componenten- (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.

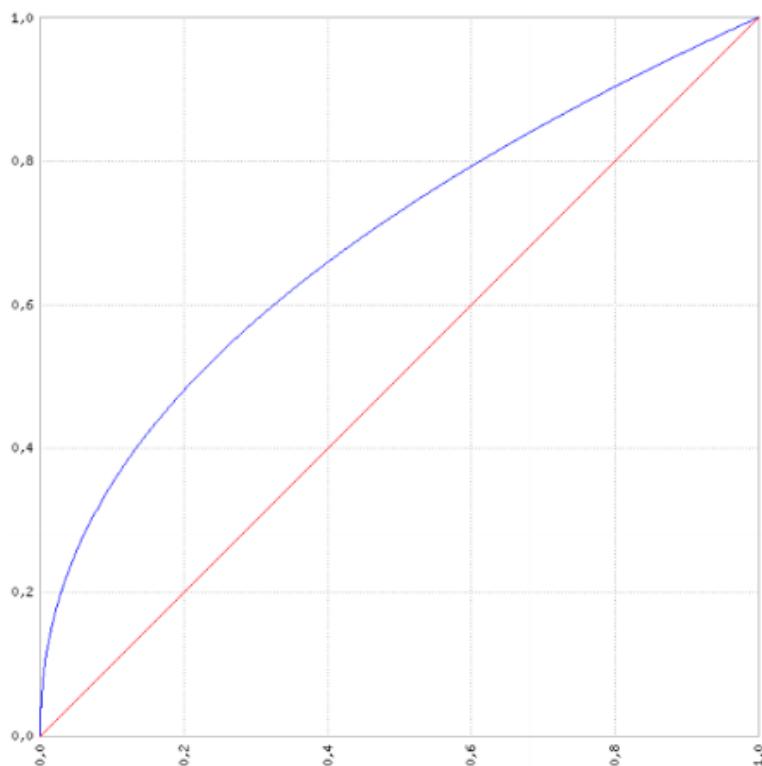
## Originalszene



## 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.

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



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

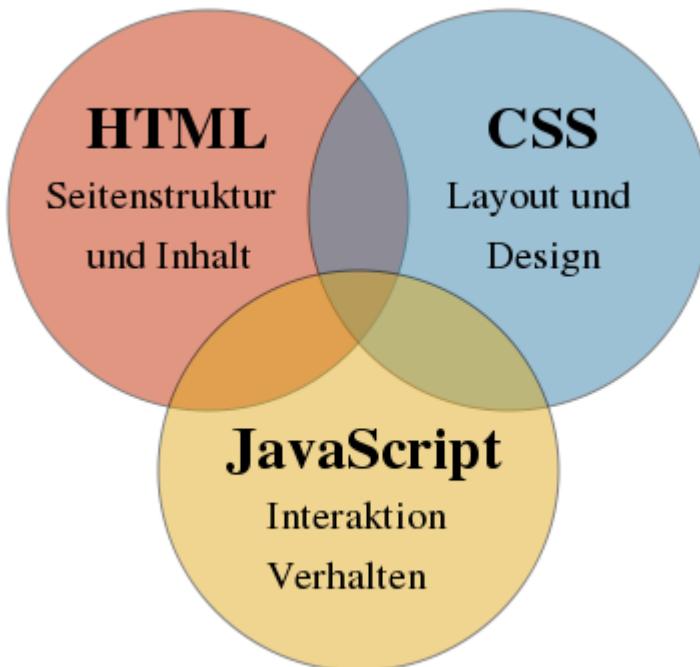


Last update: **2019/10/15 08:11**

# 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 Webauftreffs 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.

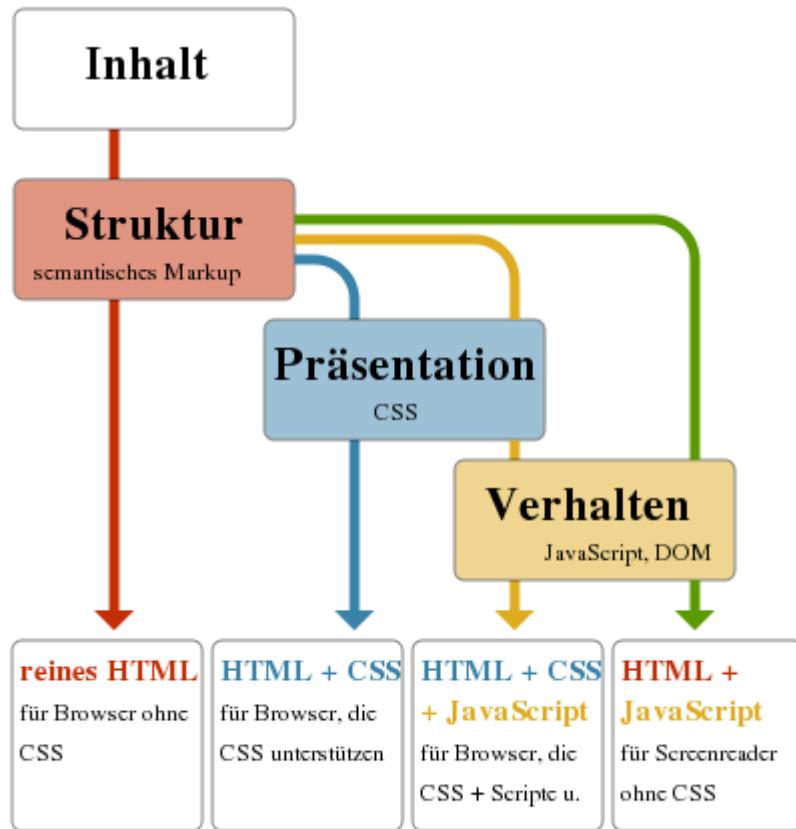
## JavaScript - Interaktivität/Verhalten

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.

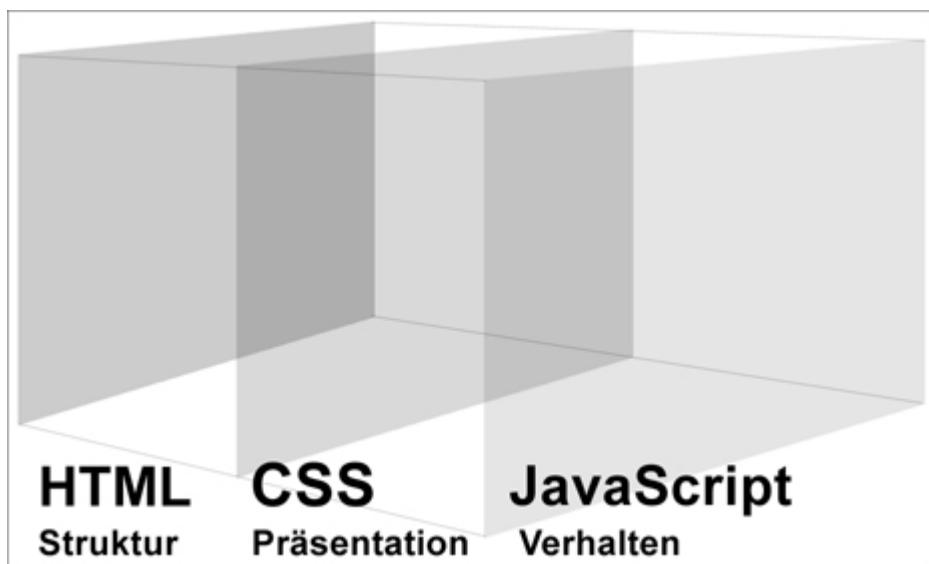
Tatsächlich liest der Browser auch zunächst diese Struktur ein, die er anschließend gemäß den CSS-Anweisungen »dekoriert;« und schließlich mit dem in Javascript beschriebenen Verhalten ausstattet. Setzen wir uns kurz mit den Grundlagen von HTML auseinander.



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

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

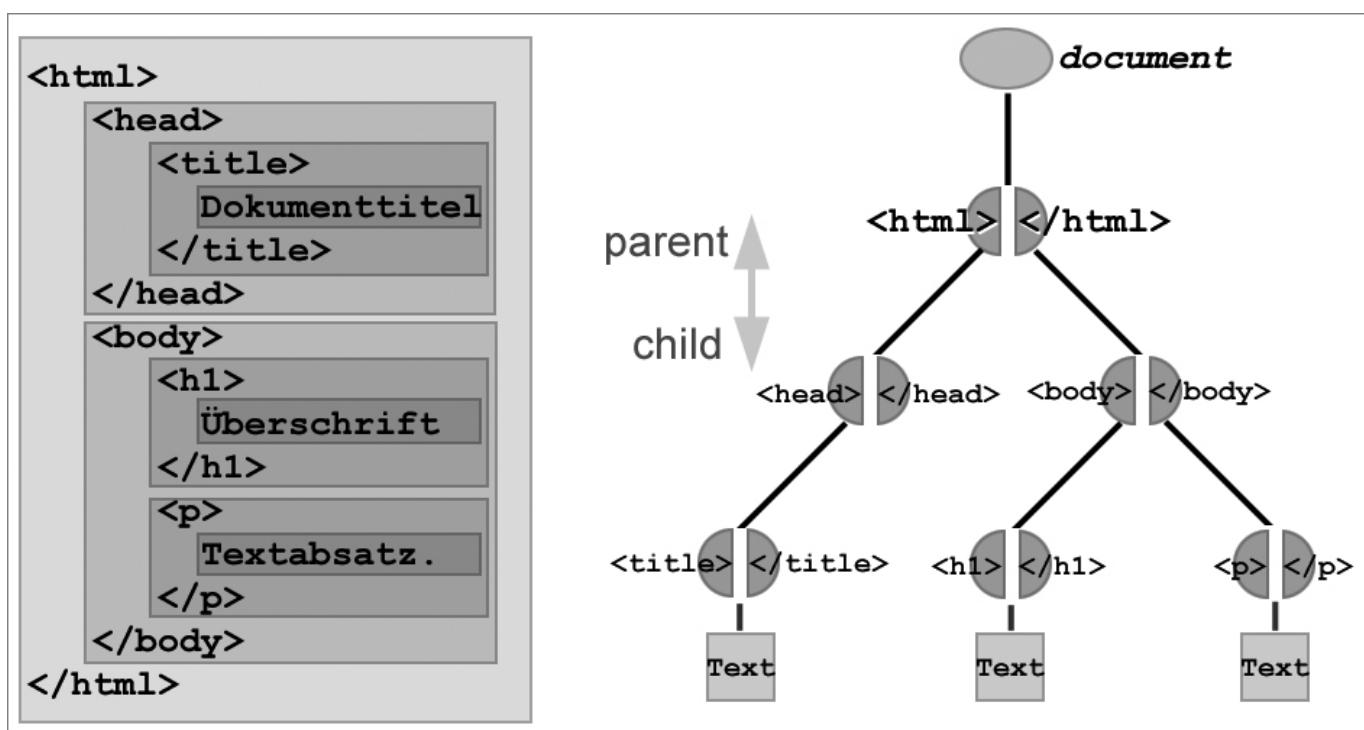
# Das Document Object Model

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-Quelltextes 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.

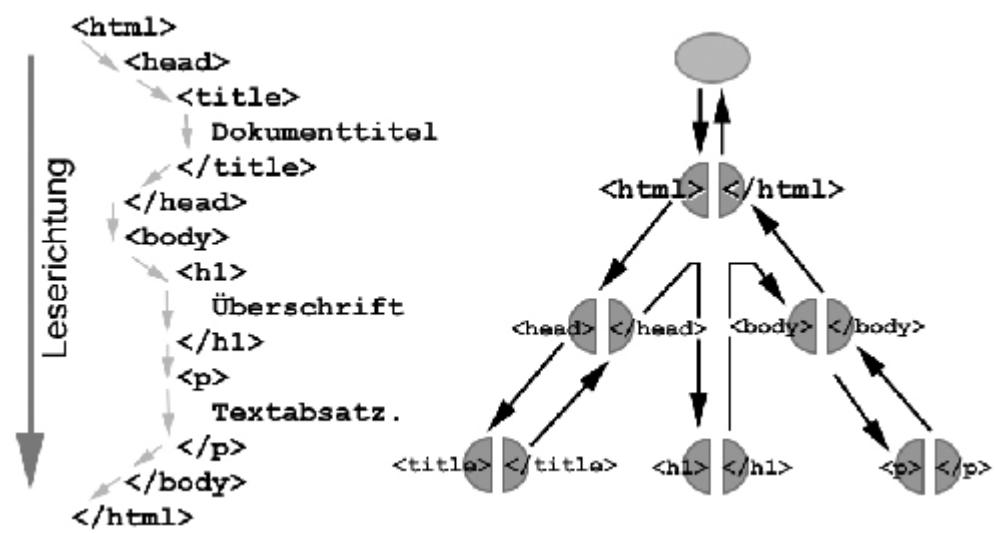
Einen solchen Dokumentknoten erzeugt der Browser beim Einlesen (Parsing) des Dokuments. Er hängt dann für jedes Element, das er in Quelltextreihenfolge antrifft, einen weiteren Knoten unten an diesen Dokumentknoten an. Das erste Element, das er antrifft, ist stets das Root-Element HTML (allgemein als `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 Quelltextes 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 Wohtgeformtheit, also die Art von regelmäßiger Struktur, die in der XMT.-Datenverarbeitung gefordert ist.

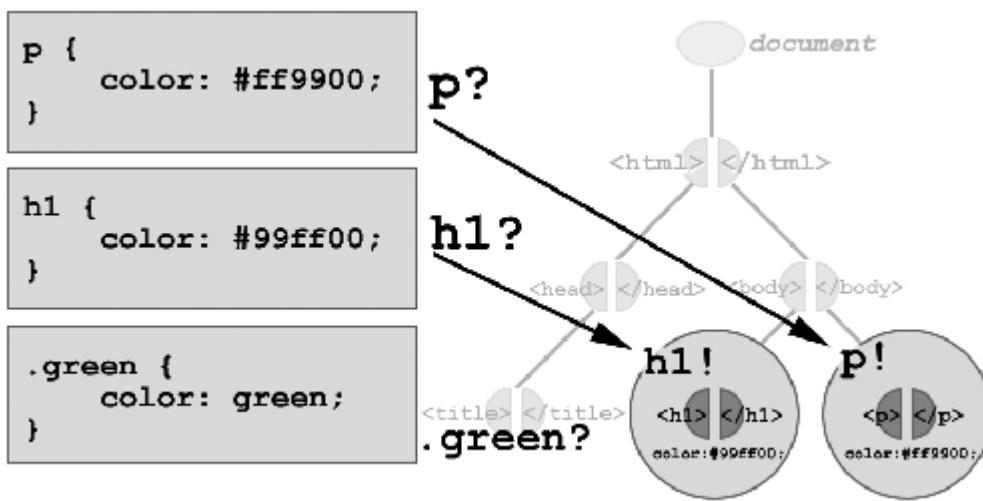
## Das »Schmücken« des DOM-Baums

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

```
body {  
    margin: 4px;  
    border: 3px dotted #000000;  
    font-family: sans-serif;  
    color: #000000;  
    background-color: #FFFFFF;  
}  
  
h1 {  
    padding: 5px;  
    margin: 10px;  
    border: 1px solid #COCOCO;  
    color: #FF0000;  
    background-color: #0000FF;  
}
```

CSS

### 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 unmengen 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-faco-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](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: #E6DACP;
    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.</span>

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](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-border-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](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](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 Ausichtung
text-indent	Texteinrückung

### Inline-Elemente

- Werden in der aktuellen Zeile (im Textfluss) dargestellt (<br /> 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 Ausrichtung) - 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](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">
```

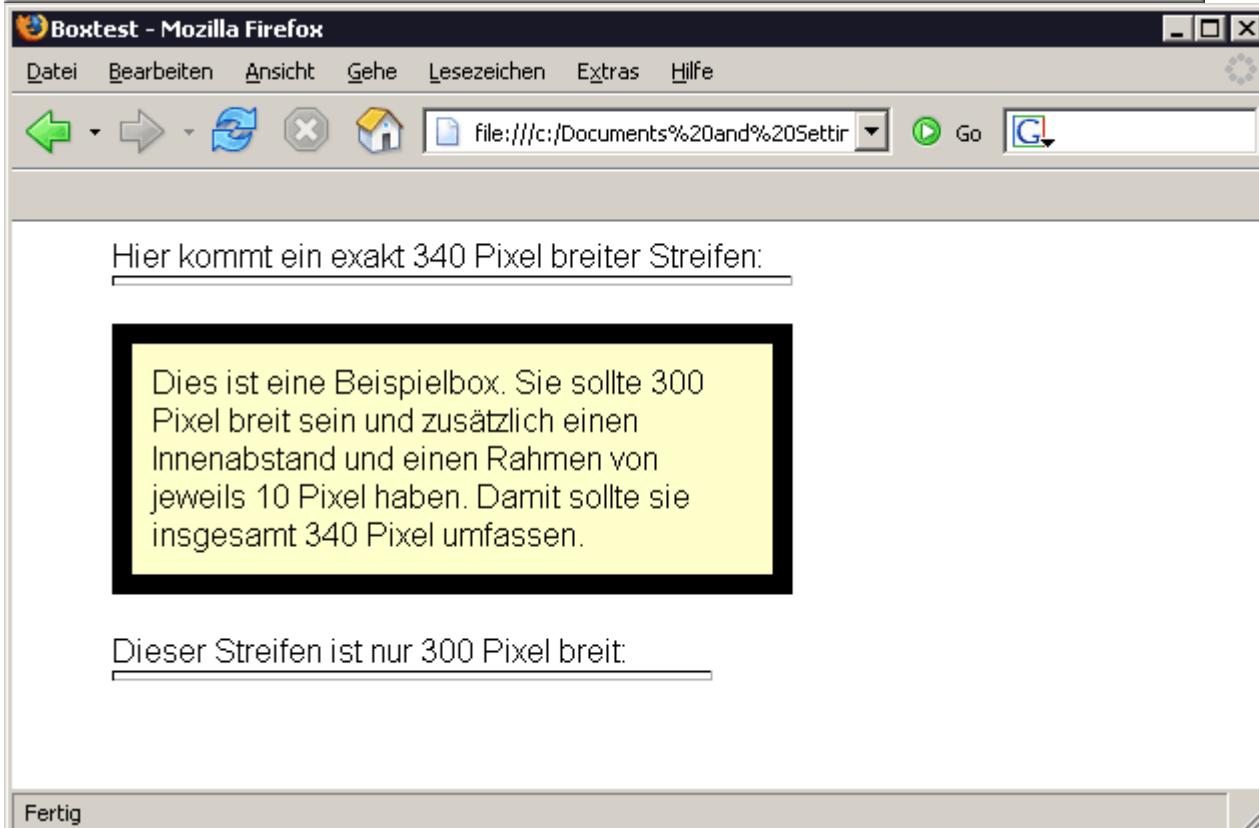
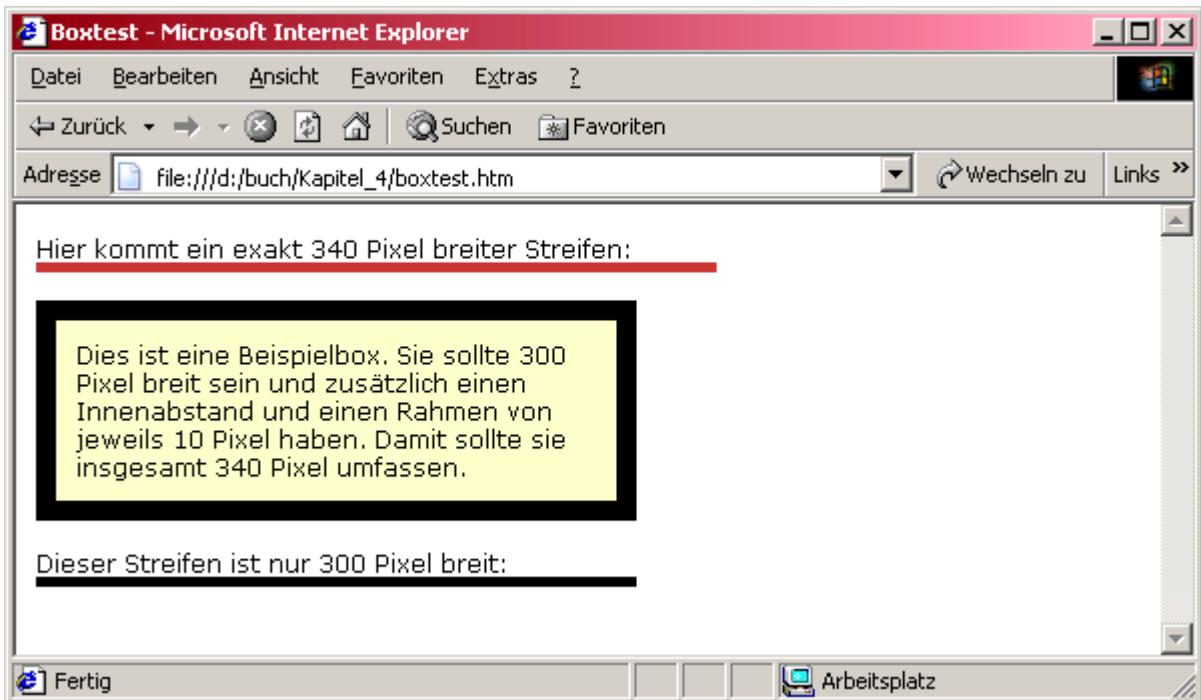
### Wichtigkeit der Dokumenttyp-Angabe

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 Modus, 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 Browsersversionen 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](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 <li>... geschrieben hat, also auf den schließenden Tag verzichtet hat, erfordert XHTML nun auch den End-Tag, also <li>...</li>.

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

mit einem Schrägstrich am Ende geschlossen werden: <img src=„LINK“ />.

Dies gilt auch für „leere Elemente“, am wichtigsten ist hier <br />, 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-Attribute 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](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](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](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

## Deklarierung

### ... in (X)HTML als Attribut

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

### ... in CSS

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

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

### auch möglich:

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

```
}
```

```
li.menupunkt {
/* 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-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 `<span>` 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](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](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-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](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](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](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**

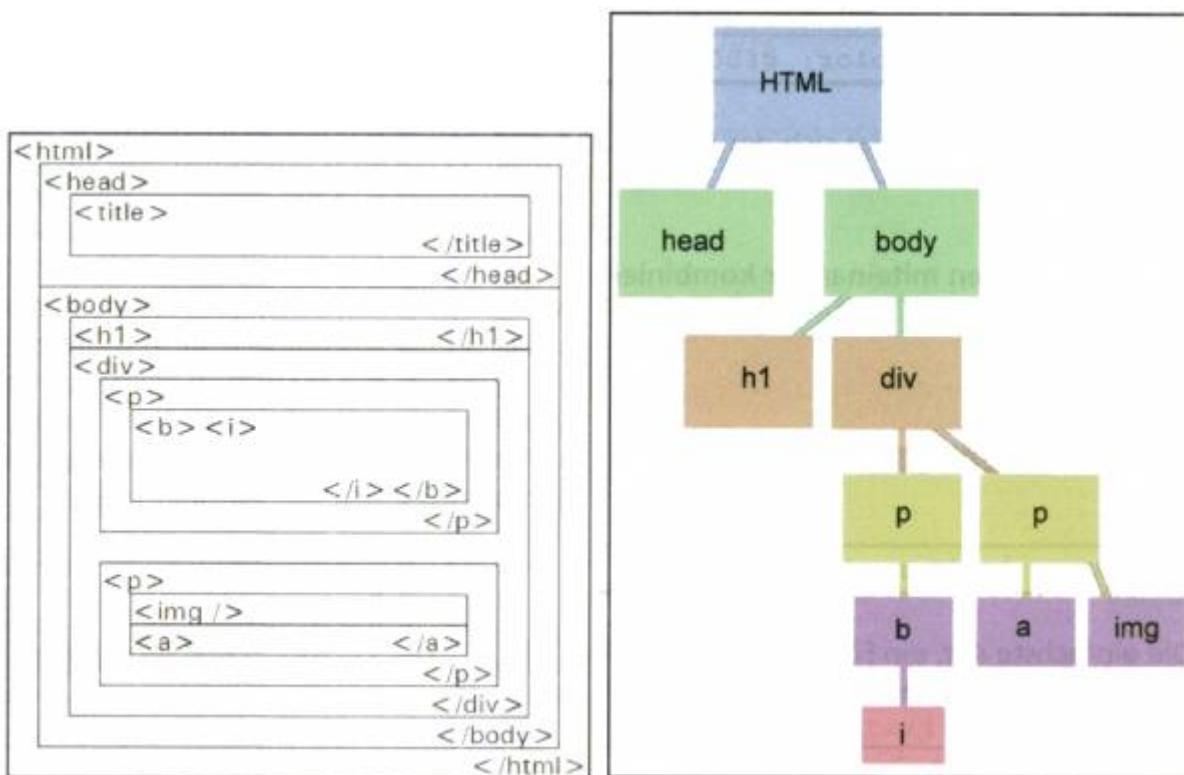
## 11.2.3.1) Vererbung

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-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 übergeordnetem Element also die Schriftgröße `font-size: 0.8em` zu, so bezieht sich eine Schriftgrößenangabe eines untergeordneten Elements auf diese `0.8em`.

**Beispiel:** Man weißt einem `<p>`-Element `font-size: 0.8em` zu, einem untergeordnetem `<span>`-Element ebenfalls `0.8em`. Dann ist die Schriftgröße des `<span>`-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-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](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">
/* <![CDATA[ */
h2 {
    font-weight: bold;
}

h3 {
    color: darkred;
}
/* ]]> */
</style>
</head>
<body>
<h1>Die wichtigste Überschrift, weil es keine weiteren Stilangaben gibt
werden die Standardeinstellungen des Browsers umgesetzt</h1>
<h2>Die zweitwichtigste Überschrift, hier gibt es eine interne Stilangabe,
diese wird auch verwendet.</h2>
<h3 style="font-color: black;">Die drittwichtigste Überschrift, hier gibt
zusätzlich zu einer internen Stilangabe auch eine Inline-Angabe, also setzt
sich diese durch.</h3>
<p class="wichtig">Beispielabsatz</p>
<p class="wichtig" style="color: white">Beispielabsatz</p>
```

```
</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-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](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](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-Deklarierung verwendet werden! */
```

## Schriftfarbe

Die Schriftfarbe eines Elements wird (wie die Farbe aller Elemente) mit der CSS-Eigenschaften **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-Eigenschaften **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. <a href="#">hier</a>
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\\_04:3\\_04\\_04\\_01](http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_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">
/* <![CDATA[ */
body {
  font-family: sans-serif;
  background-color: #FFF;
  color: #306;
}
ul {
```

```
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](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">
/* <![CDATA[ */
body {background-color: #333; }
a:link { color: red; }
a:visited { color: green; }
a:hover { color: blue; }
a:focus { color: white; }
a:active { color: yellow; }
/* ]]> */
</style>
</head>
```

```
<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**



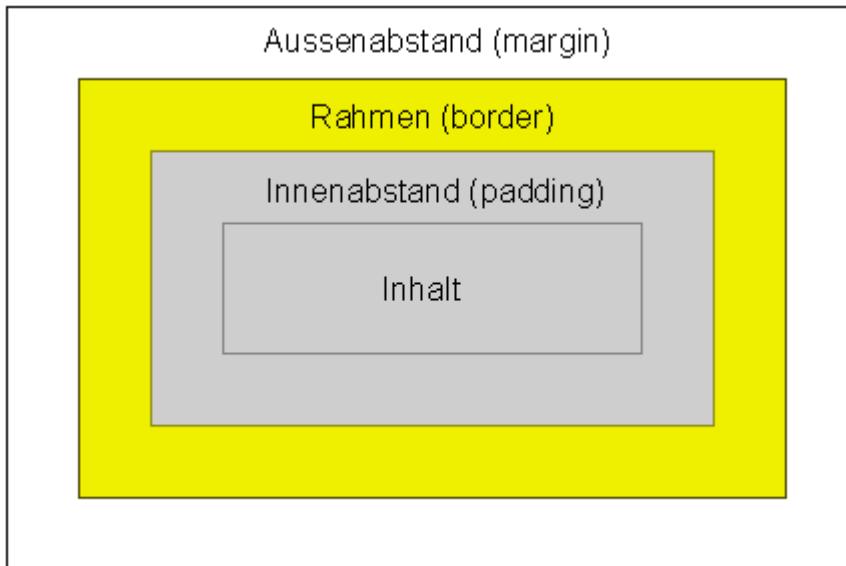
Permanent link:  
[http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi\\_201920:3:3\\_04:3\\_04\\_04:3\\_04\\_04\\_03](http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_04:3_04_04_03)

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">
/* <![CDATA[ */
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;
}
/* ]]> */
</style>
</head>
<body>
<div id="box">
    <div id="inhalt">Inhaltsbereich</div>
</div>

</body>
</html>
```

## Links

- [CSS4You: Informationen über das Boxmodell](#)
- [CSS4You: Eigenschaften für Rahmen](#)
- [CSS4You: Eigenschaften für Außenabstand](#)
- [CSS4You: Eigenschaften für Innenabstand](#)

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](http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_04:3_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-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](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 cellspacing="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;lft
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](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.

### Statische Positionierung

Die statische Positionierung ist die häufigst-angwendete, 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-Type" content="text/css" />
    <meta http-equiv="content-language" content="de" />
    <title>TITEL</title>

<style type="text/css">
/* <![CDATA[ */
<!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>statische Positionierung</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<style type="text/css">
/* <![CDATA[ */
div {
    width: 12em;
    color: #00000CC;
    border: dotted 1px;
    padding: 1em;
}
.box1 {
    background-color: #FFCC00;
    position:static;
}
.box2 {
    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 Element 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">
/* <![CDATA[ */
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;
}
/* ]]> */
</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">
/* <![CDATA[ */
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;
}
/* ]]> */

```

```
</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">
/* <![CDATA[ */
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 {
    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, consectetuer 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, consectetuer 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, consectetuer 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, consectetuer 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, consectetuer 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">
/* <![CDATA[ */
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;
}
/* ]]> */
</style>
</head>
<body>
<div class="box1">Erste Box</div>
<div class="box2">Zweite Box</div>
<div class="box3">Dritte Box</div>
</body>
</html>
```

## Durchsichtigkeit (Opacity)

Ebenfalls interessant ist die Eigenschaft **opacity**: 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.

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



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

## 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, weißt 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">
/* <![CDATA[ */
* {
    padding: 0;
    margin: 0;
}
body {
    font: 100.01% sans-serif;
}
div {
    background-color: #EBE6E0;
    margin: 10px;
}
.rechts {
    float: right;
}
.links {
    float: left;
}

/* ]]> */
</style>
```

```

</head>
<body>
<div><p>Lorem ipsum dolor sit amet, consectetuer 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, consectetuer 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: Umfleßen beenden

Das Problem ist nun, dass das Bild nicht nur von dem einen Absatz umflossen wird, sondern von allen folgenden Elementen. Das Umfleßen 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 umfleßen 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">
/* <![CDATA[ */
* {
  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: #E6E6E0;
    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">
/* <![CDATA[ */
* {
    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: #E6E6E0;
    margin: 10px;
    overflow: hidden;
    width: 900px;
}
hr {
    clear: left;
}
/* ]]> */
</style>
```

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\\_08](http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_04:3_04_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.
		<b>action:</b> Zielfile des Formulars beim Absenden, <b>method:</b> Methode der Datenübertragung (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. <b>for:</b> Muss dem Attributwert „id“ des input-Felds entsprechen
<input />	type, name, id, size	Eingabeelement <b>type:</b> Feldtyp (z.B. text), <b>name:</b> eindeutige Bezeichnung für PHP, <b>id:</b> eindeutige Bezeichnung für HTML/CSS/JS, <b>size:</b> 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.

Anzeigbeispiel - 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\\_04\\_09](http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_04:3_04_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](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 Erlkönig.

- 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 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 feiner 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 feiner Knabe du mit mir geh'n?  
Meine Töchter sollen dich warten schön,  
Meine Töchter führen den nächtlichen Reih'n  
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](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](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](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>Übung - 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>Übungstabellle</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](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)



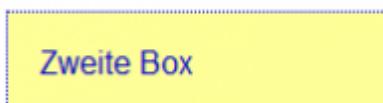
Last update: **2019/11/17 16:12**

### 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:



Zwischentext

(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"
alink="#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](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, consectetuer 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.

A small image of a kitten sitting in grass, used as a placeholder for a floating element.

A large image of a kitten sitting in grass, used as a placeholder for a floating element.

A large image of a kitten sitting in grass, used as a placeholder for a floating element.

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](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](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](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 Dokumenteneigenschaften, 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 (<ol> Element), indem du
- Grossbuchstaben verwendest (list-style).
  - die Textfarbe wechselt.
  - den Text fett und kursiv setzt. (font-weight, font-style).
6. Verändere die Brokerliste (<ul> 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 Dokumenteneigenschaften, 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](http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6bi_201920:3:3_04:3_04_05:3_04_05_02)

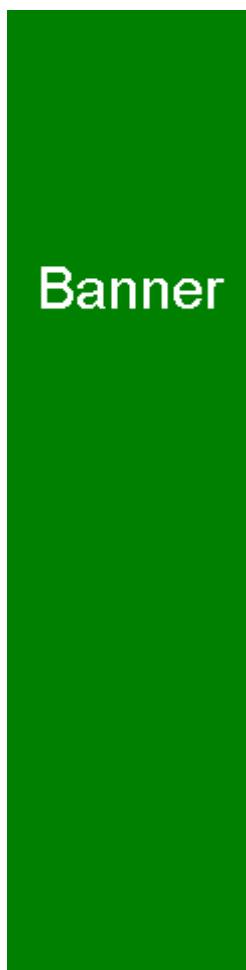


Last update: **2019/11/17 16:15**

## 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 <a href=„#“>Impressum</a> geben
- Für den Fall das 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 120x480 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.

Banner

## Vorteile bei XY

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

## Jetzt anmelden

### Daten eingeben

Username:

E-Mail:

Passwort:

Passwort wiederholen:

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 L&ouml;sung f&uuml;r die Kommunikation in Ihrer Firma.<br>
Wir bieten Ihnen ein schnelles, sicheres und zuverl&auml;ssiges Sprach- und
Textkommunikationssystem,<br>
```

```
welches über mehrere Netzwerke seine Vorteile behält.<br>
<h2>Vorteile bei XY</h2>
<ul>
  <li>Jederzeit verfügbar</li>
  <li>Schnell</li>
  <li>Sicher</li>
  <li>Zuverlässigkeit</li>
  <li>Einzigartige klare Sprachqualität</li>
  <li>Keine Authentifizierungsprobleme dank Anbindung 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="password" size="30"
maxlength="30"><br>
Passwort wiederholen: <input type="Password" name="password1" size="30"
maxlength="30"><br>
<input type="Submit" name="absenden" value="Absenden" type="submit"> <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](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 Menüpunkt 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!

**Startseite (h1)**

**Logo**

**MENU 1**

- Hyperlink 1
- Hyperlink 2
- Hyperlink 3
- Hyperlink 4
- Hyperlink 5

**MENU 2**

- Hyperlink 1
- Hyperlink 2
- Hyperlink 3
- Hyperlink 4
- Hyperlink 5

**MENU 3**

- Hyperlink 1
- Hyperlink 2
- Hyperlink 3
- Hyperlink 4
- Hyperlink 5

**Blindtext von www.lorem ipsum.de (h2-Überschrift)**

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet, consetetur adipisciing 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. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Lorem ipsum dolor sit amet, consetetur adipisciing 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.

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](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](http://webmasterpro.de))
- CSS auf [4webmaster.de](http://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](http://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](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**