

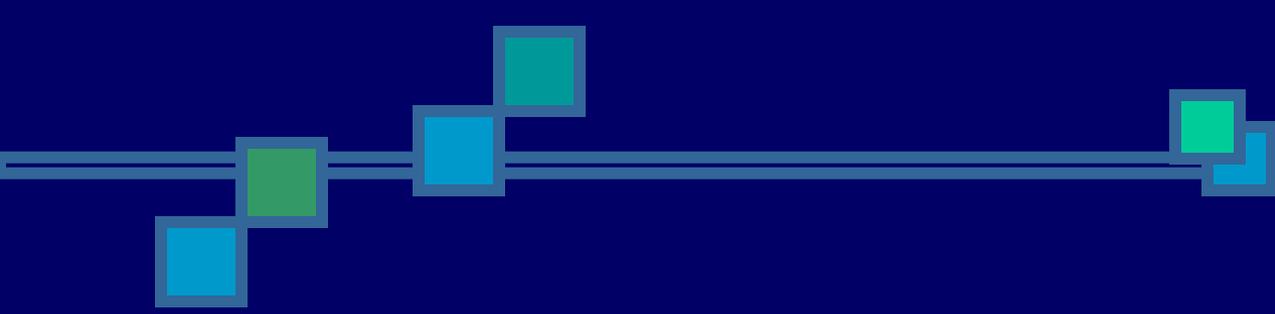
Musik und Computer

(Version SoS 2014)

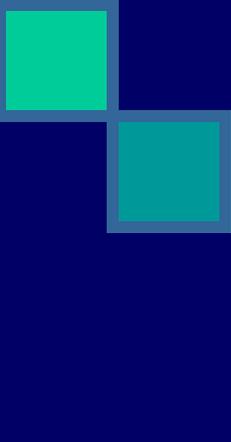


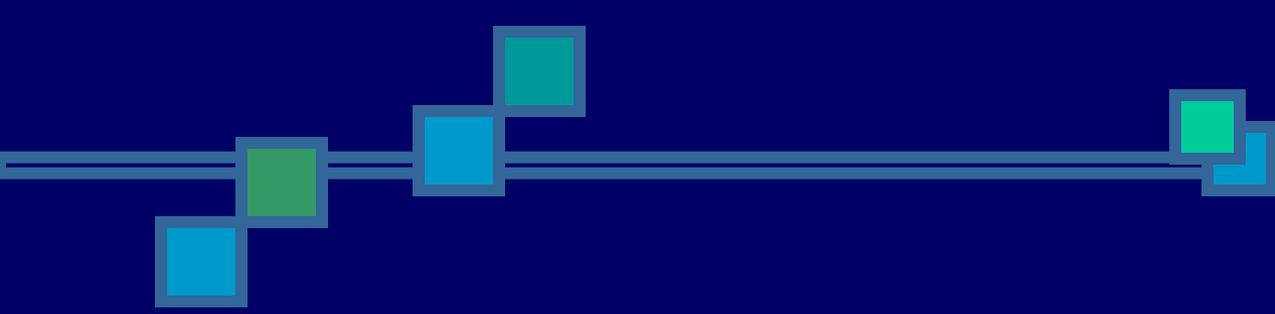
Prof. Dr. K. Huckert

Projektgruppe Praxisorientierte Informatik,
HTW des Saarlandes



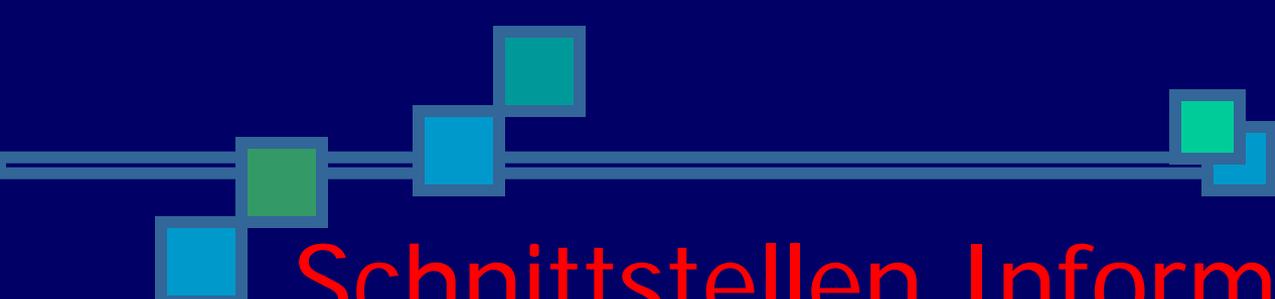
Gliederung der Vorlesung:

- Einsatzgebiete Computer in der Musik
 - Historie
 - Grundlagenwissen Musik
 - Grundlagen von Audio, MP3 und Midi
 - Software für Musik
 - Notationssoftware
 - Sequenzer
 - Arrangiersoftware
- 

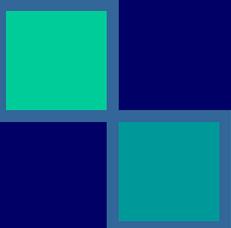


Zielsetzungen der Vorlesung:

- Einsatzbereiche von Software im Musikbereich darstellen
 - Kennenlernen von einigen Software-Anwendungen
 - Schnittstellen Informatik und Musik aufzeigen
- 



Schnittstellen Informatik und Musik



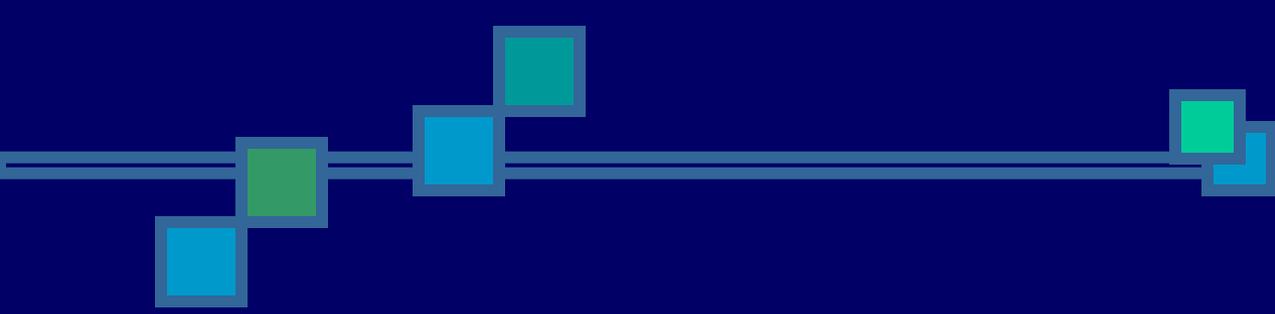
■ Beispiele

- Notation und XML
 - Akkorde und Logik
 - Akkorde und Datenbanken
 - Musikalische Analyse und Pattern Matching
 - Improvisation und „künstliche Intelligenz“
 - Komponieren und Regeln
 - Audio und Kompressionsverfahren
- 

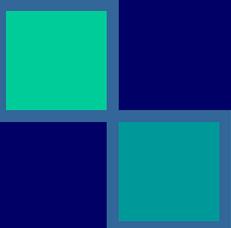


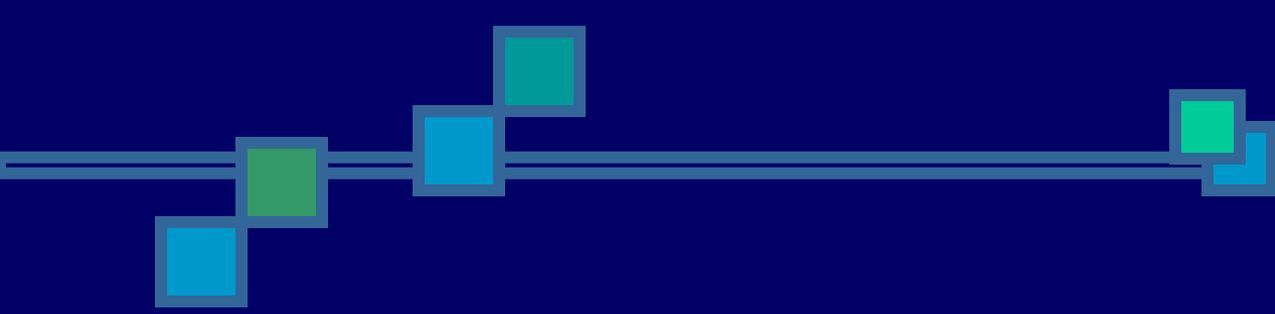
Schnittstellen Informatik und Musik

- Beispiele
 - MP3 und Fouriertransformationen
 - Musik und Digitalisierung
 - MIDI und XML
 - Instrumente und VST-Technologie
 - Musizieren im Orchester vs. Künstliches Orchester
- 

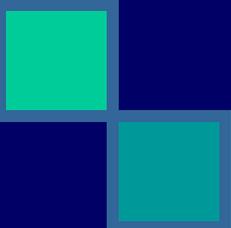


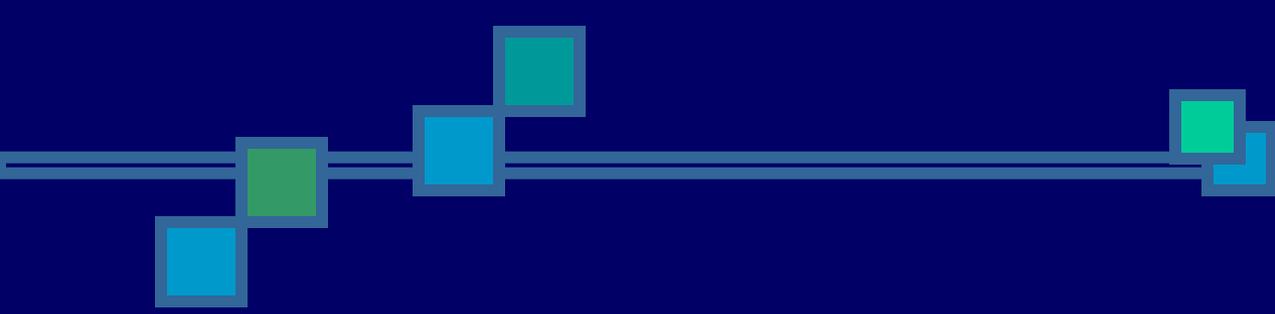
Die heutige Situation:

- 
- Computer durchdringt alle Arbeitsbereiche
 - Musikbereich
 - Üben
 - Komponieren
 - Arrangieren
 - Virtuelle Instrumente(VST)
 - Aufnehmen, Mischen und Mastern
 - Verwaltung von Datenbeständen
 - Einsatz im Internet, CD-ROM, DVD
- 

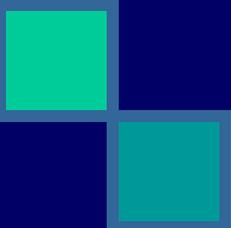


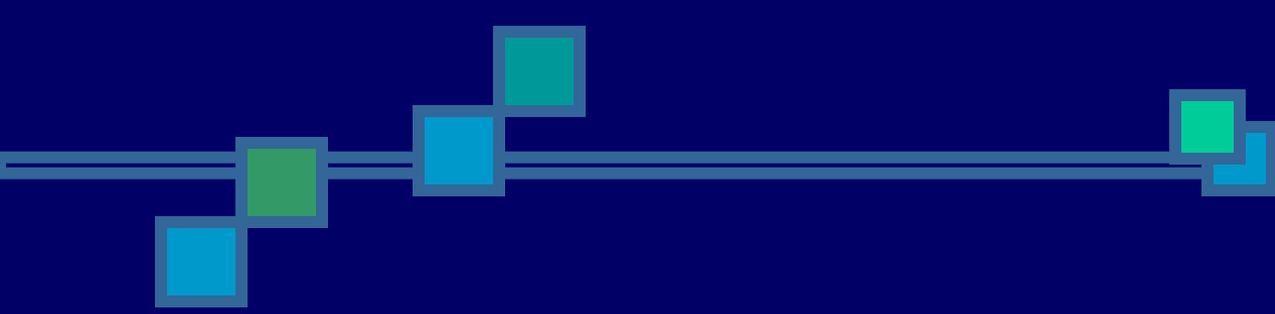
Wichtige Meilensteine:

- 
- Klavier-/Orgelsteuerung über Lochstreifen
 - Trautonium, Mellotron
 - Karl Heinz Stockhausen
 - Musikgruppen wie Can, Kraftwerk, Neu!
 - Entwicklung von Synthesizern
 - Midi-Format
 - Entwicklung von Soundkarten für PCs
- 

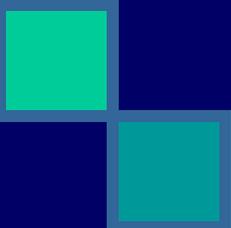


Grundlagenwissen Musik

- 
- Vgl. Ausarbeitung Musik-Grundlagen Neuschwander/Huckert
 - Ausarbeitung auf nichtöffentlichem Server
 - Gliederung auf der nächsten Folie
- 

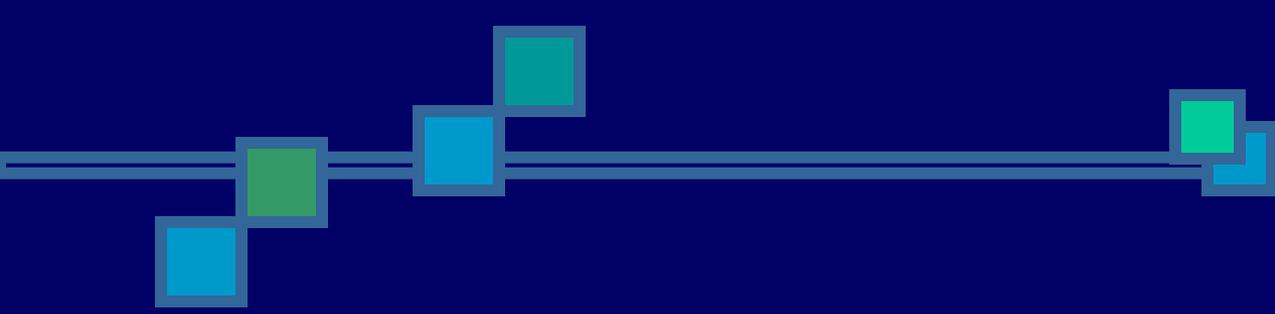


Grundlagenwissen Musik

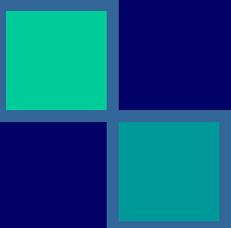


■ Gliederung

1. Musikdarstellung im Notensystem
 - 1.1 Einführung
 - 1.2 Noten Grundlagen, Notenwerte, Takt, Notenschlüssel
 - 1.3 Vortragsbezeichnungen
 - 1.4 Die Notenvorzeichen
 - 1.5 Intervalle
 - 1.6 Tonarten bzw. Tonleitern
 - 1.6.1 Die Dur-Tonleitern
 - 1.6.2 Die Moll-Tonleitern
 - 1.6.3 Kirchentonarten
 - 1.7 Akkorde
 - 1.8 Jazz-Akkorde
 - 1.9 Erweiterungen der Jazz-Akkorde
 - 1.10 II-V-I-Verbindungen, I-VI-II-V-Verbindungen
 - 1.11 Harmonieanalyse



Grundlagenwissen Musik

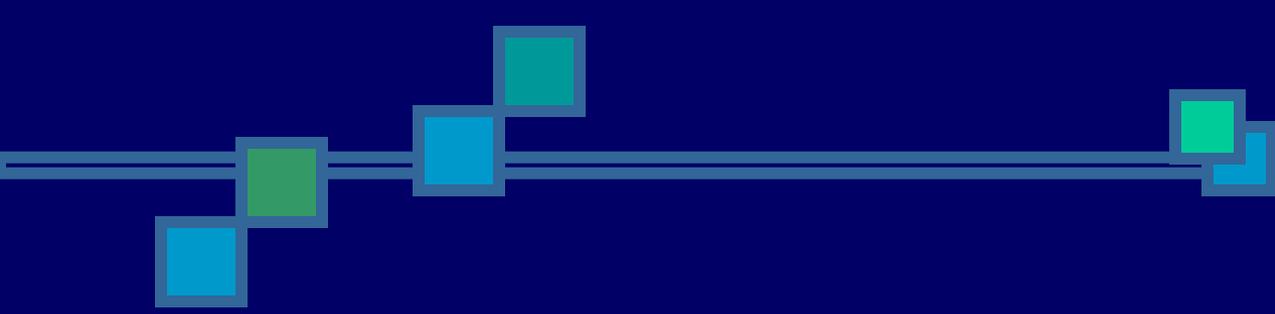


Notationssoftware

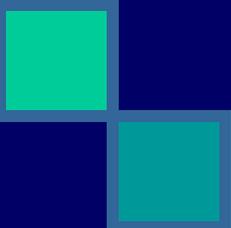
Praktikum mit Finale PrintMusic!

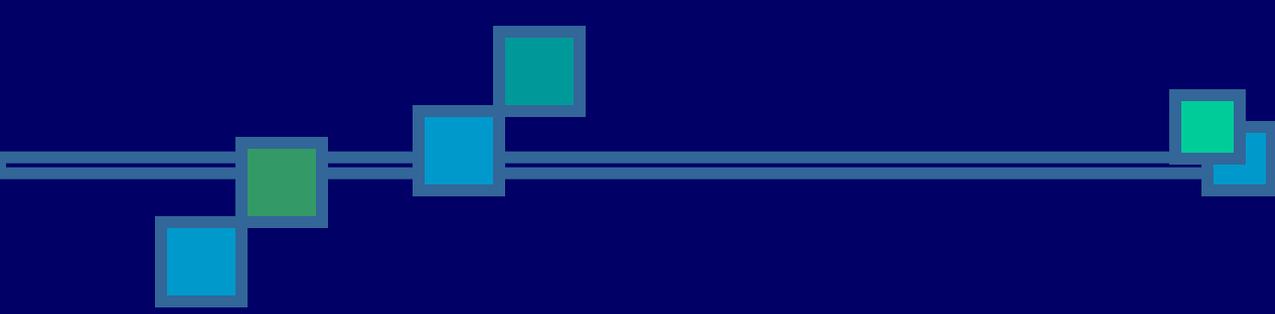


Fallstudie: Capella

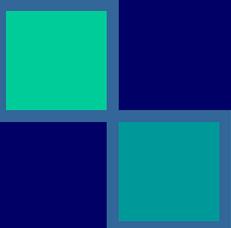


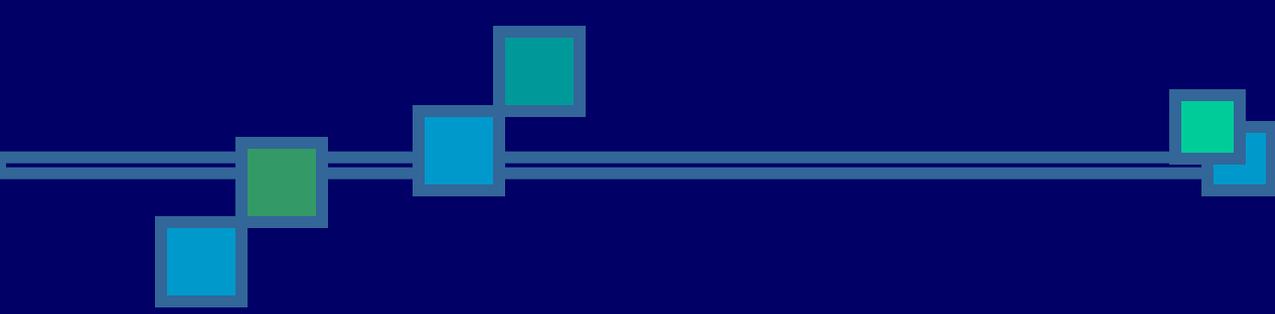
Praktikum 1

- 
- 
- Erfassen Sie das Musikstück "On the banks of the Ohio" mit Finale Print Music!
 - Interpretieren Sie die Akkorde über dem Musikstück
 - Transponieren Sie das Stück von C-Dur nach F- und G-Dur.
 - Erzeugen Sie eine Midi-Datei in C-Dur.
 - Ordnen Sie den Text dem Musikstück in C-Dur zu.

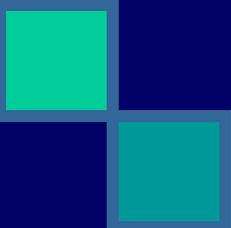


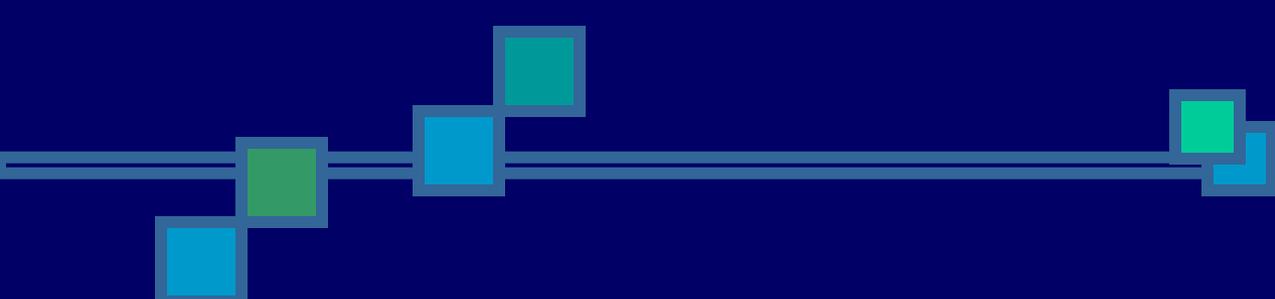
Praktikum 1

- 
- Erfassen Sie das Stück „On the Sunny side of the street“
 - Erzwingen Sie eine 4 taktige Ausrichtung pro Zeile
 - Transponieren Sie aufwärts nach G-Dur
 - Erzeugen sie einen entsprechenden Midi-File.
 - Untersuchen Sie die .MUS-Datei mit einem Hex-Editor, versuchen Sie den Inhalt zu interpretieren!
- 



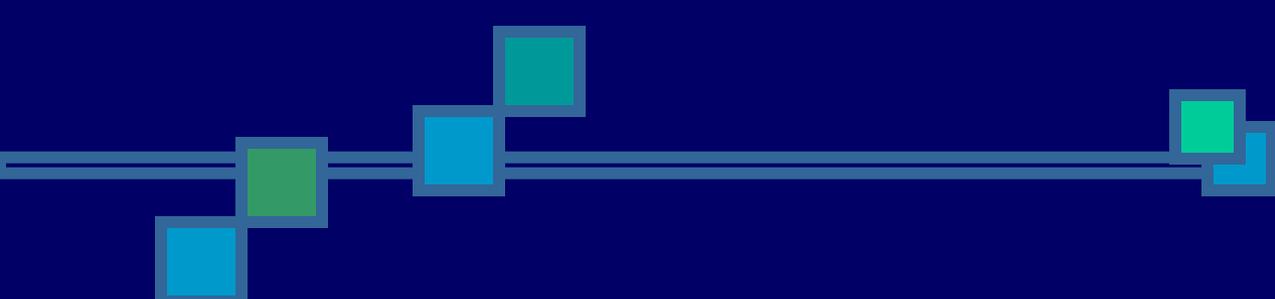
Praktikum 1

- 
- Laden Sie den Midi-File "Blue Bossa"
 - Schreiben Sie mit Finale Print Music die einzelnen Stimmen heraus
 - Wandeln Sie die Drum-Darstellung in eine rhythmische Notation um
 - Erstellen Sie nur von der Melodie einen Midi-File
- 



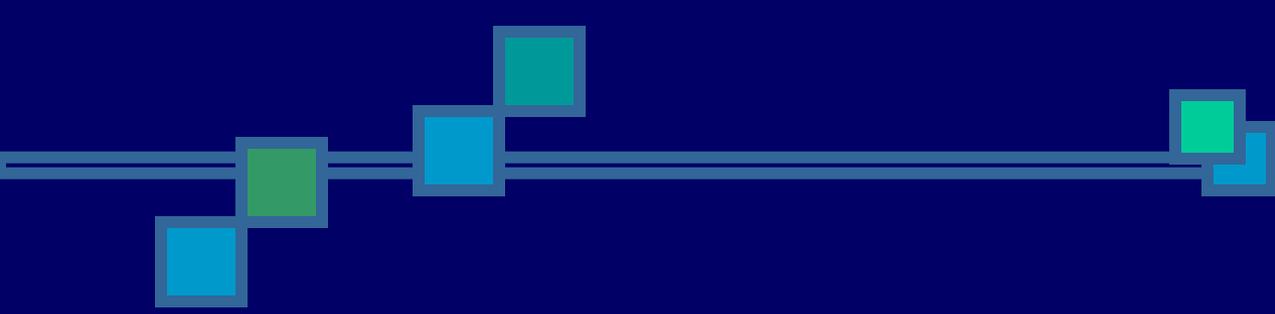
Grundfunktionalität Notationsprogramme (1)

- Erfassung von Titel, Komponist, Stimme, Tempo, Ton-/Taktart
 - Erfassung von Noten in verschiedenen Systemen
 - Artikulation, Vortragsbezeichnungen
 - Erfassung von Texten
 - Akkordzuweisungen
- 

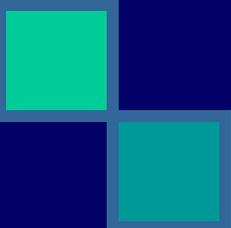


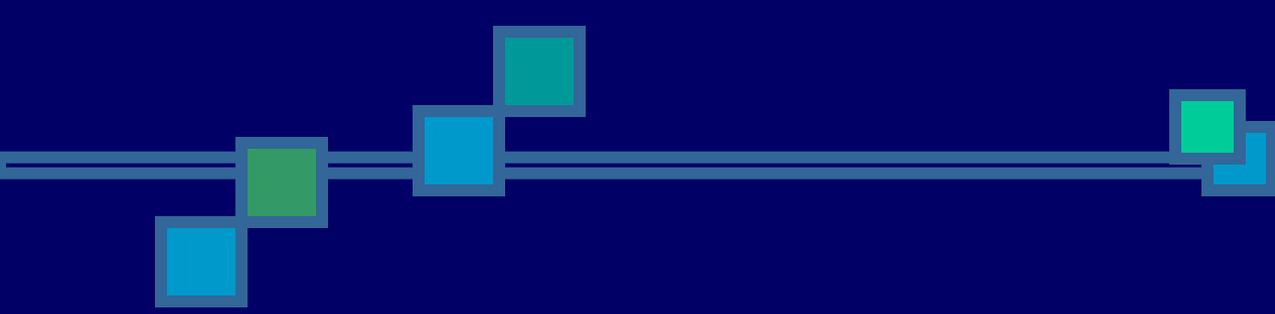
Grundfunktionalität Notationsprogramme (2)

- Wiederholungen
 - Abspielen erfasster Noten
 - Transposition
 - Abspeichern im Midi-Format
 - Abspeichern im XML-Format
 - Scannen von Notenblättern
 - Eingabe über Midi-Schnittstellen
- 



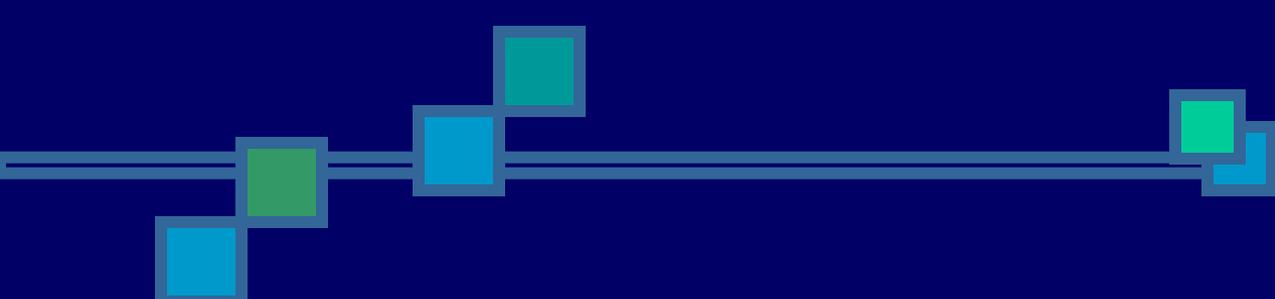
Notationsprogramme

- 
- Finale (kostenlos Finale Notepad)
 - Sibelius
 - Capella
 - Forte (kostenlos Forte Freeware)
- 



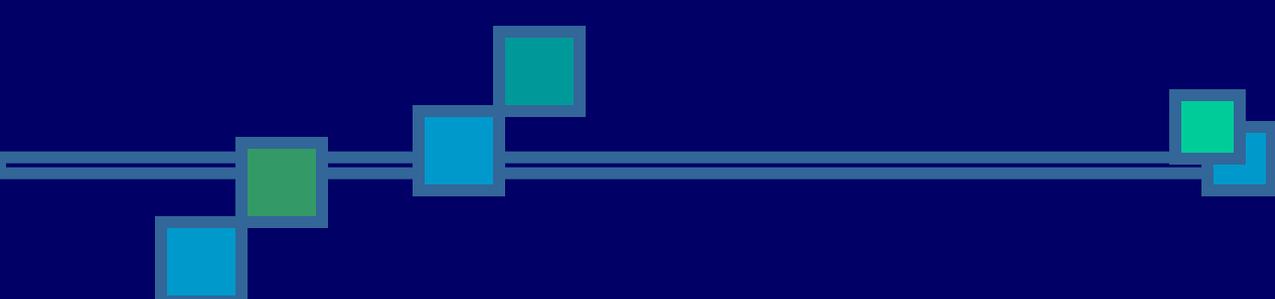
Notationsprogramme

- Kurzer Vergleich von drei Notationsprogrammen unter
 - <http://www.nmz.de/studicard/inhalte/notensatz.shtml>
- 



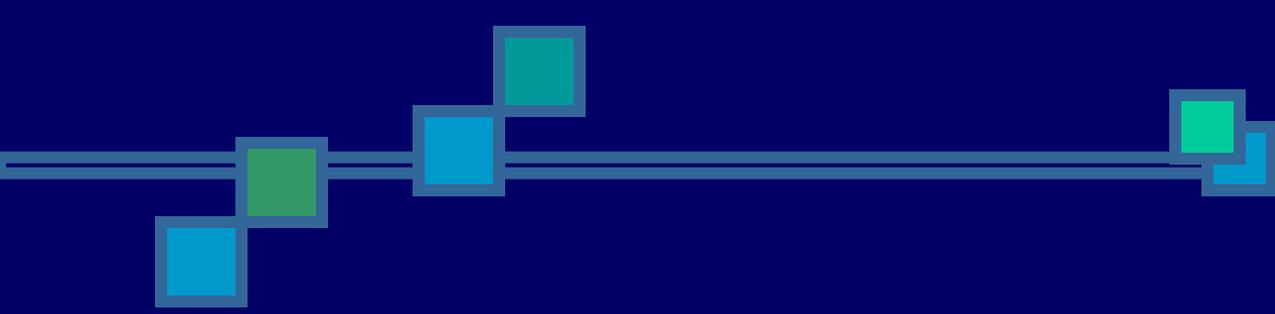
Problematik Notationsprogramme

- Abspeicherung in diversen Formaten
 - nicht immer unbedingt einfach
 - hohe Fehlerraten in Scan-Schnittstellen
(siehe auch Testbericht c´t)
 - Updates nicht immer kompatibel
- 

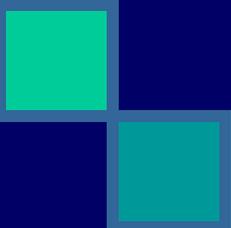
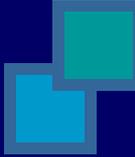


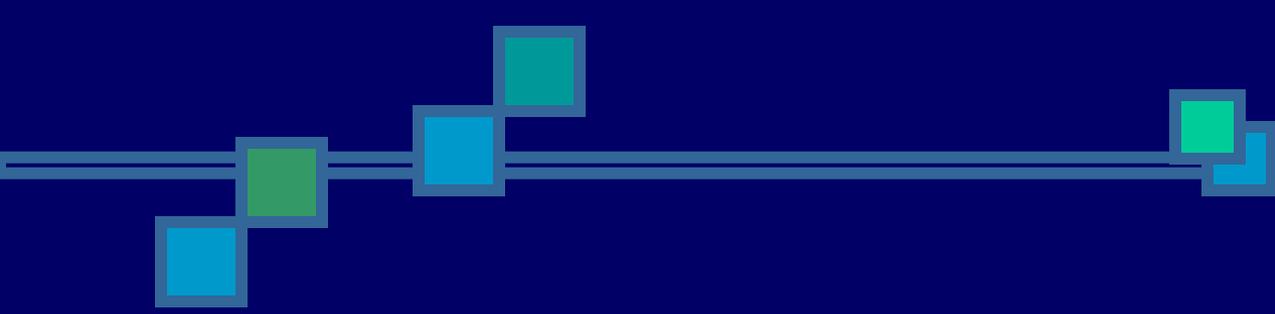
Notationsprogramme und Informatik

- XML-Format
 - Soundgenerierung
 - Automatische Akkordermittlung
 - Darstellung in unterschiedlichen Notationssystemen (Notenschrift, Rhythmus)
 - Automatische Analyse (z.B. 1-6-2-5-Verbindungen oder 2-5-1-Verbindungen)
- 

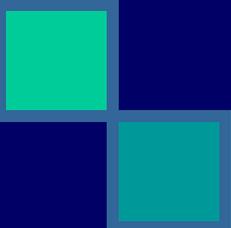


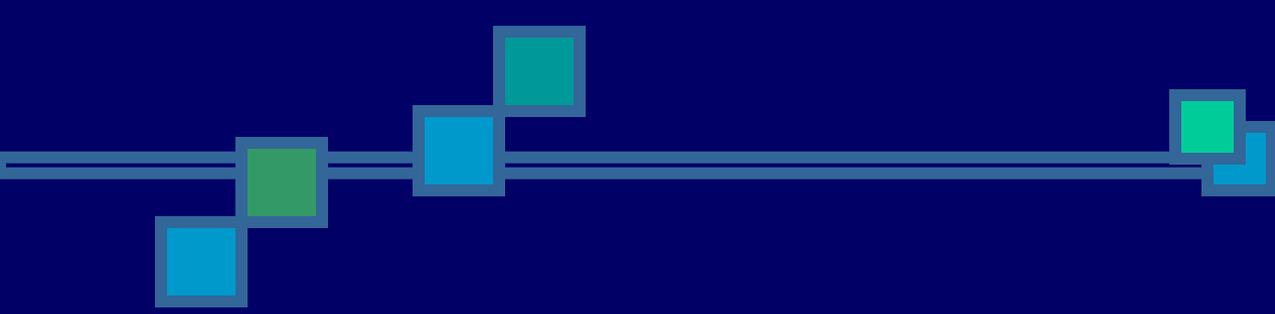
Notationsformate (1)

- 
- Welche Datenformate kommen für die Übertragung von Notationen in Frage?
 - Proprietäre Formate der jeweiligen Hersteller (siehe Print MUSIC!)
 - NIFF (Notation Interchange File Format)
 - erstes offenes Standardformat für Musiknotation
 - professionelle Unternehmen an Entwicklung beteiligt
 - Schwerpunkt: universell unterstütztes, erweiterbares, flexibles und trotzdem kompaktes Dateiformat
 - Verbreitung allerdings bis heute eher gering
- 



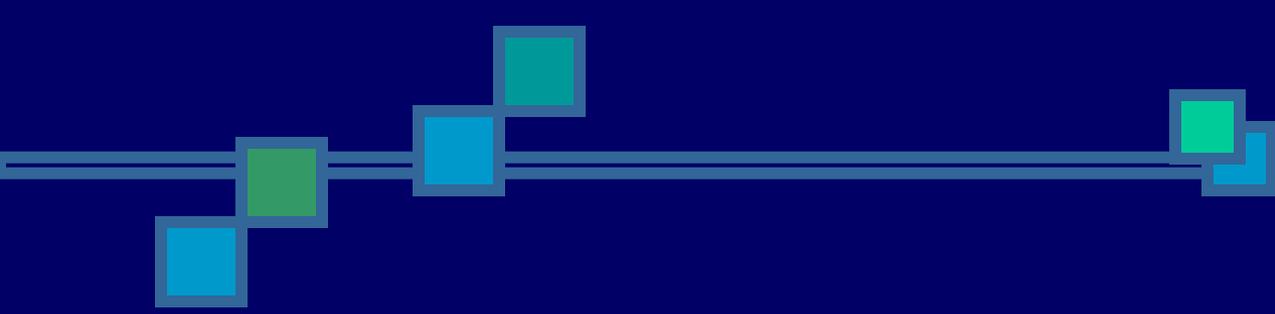
Notationsformate (2)

- 
- Welche Datenformate kommen für die Übertragung von Notationen in Frage?
 - SMDL (Standard Music Description Language)
 - SGML als Grundlage, sehr hohe Komplexität
 - Kompatibilität zu XML nicht vollständig gewährleistet
 - GUIDO Music Notation Language
 - ASCII-basierte Notationssprache mit starker Ähnlichkeit zu TeX
 - transportables, einfaches und robustes Datenformat
 - **vgl. Aufsatz von Edgar und Klaus Huckert zur Notationsproblematik unter www.jazzy-listening.de**
- 

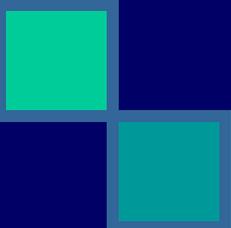


Lösung aus Informatik-Sicht

- Auswahl verfügbarer XML-Standards:
 - MusicXML
 - capXML
 - MusiXML
 - MNML (Music Notation Markup Language)
 - EMNML (Extensible Music Notation Markup Language)
 - XScore (eXtensible Score Language)
 - MML (Music Markup Language)
 - MEI (Music Encoding Initiative)
- 

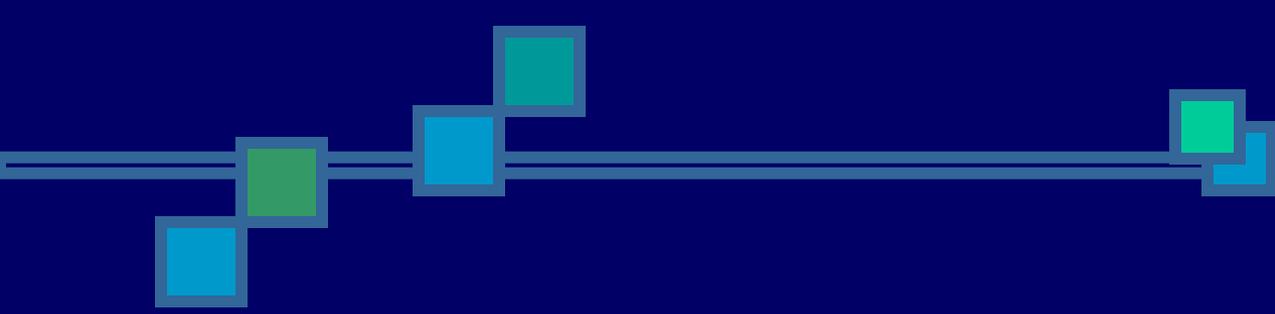


XML

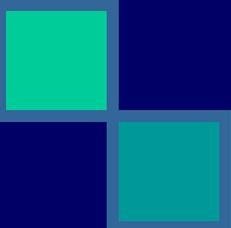


■ Problematik MIDI

- Binäres Datenformat, nur für Spezialisten lesbar
 - Variable Interpretation durch unterschiedliche Hersteller
 - Ungenaue Darstellung MIDI zu Notationsformat (Notendarstellung, Vortragsbezeichnungen etc.)
 - Keine Unterstützung Layout
 - Keine Unterstützung Akkordspeicherung
- 



XML



- **Vorteile XML**

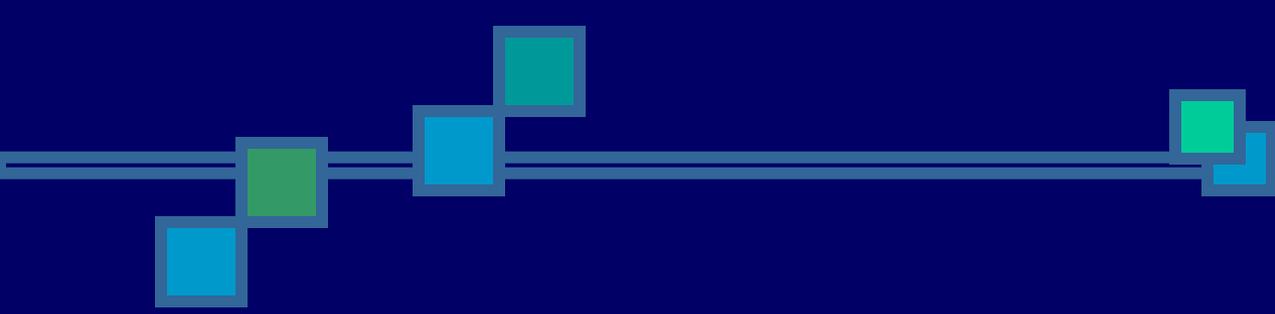
- Repräsentation komplexer und strukturierter Datenstrukturen realisierbar**

- einfaches Datenformat (ASCII), sehr stabil und robust**

- Speicherung von Meta-Daten ermöglicht Interpretation des Formates auch nach Jahrzehnten**

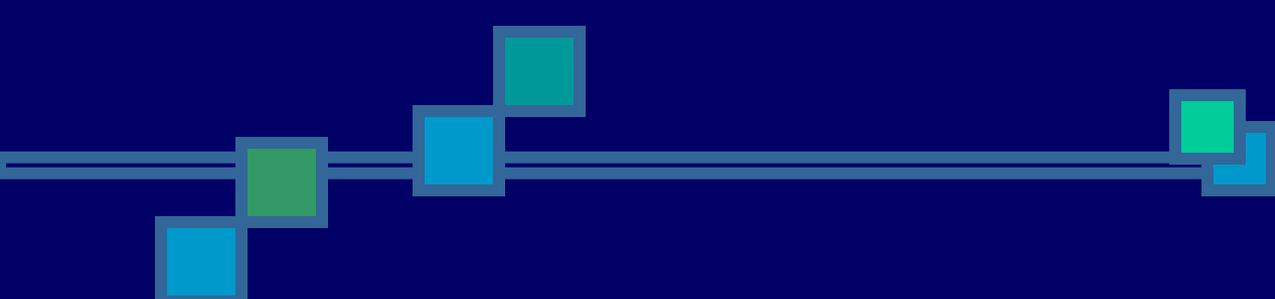
- XML ist für Menschen lesbar und verständlich**

- Cross-Media-Publishing (Druck, Webseite, MIDI, CD-ROM, Wap-Handy, Klingeltöne)**

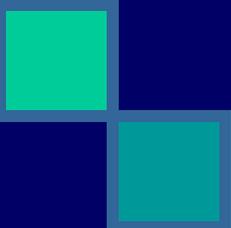


XML

- Wichtige Internetquellen
 - www.musiknotation.de
 - www.music-notation.info
 - www.musicXml.org
 - www.recordare.com
 - www.capella.de
- 

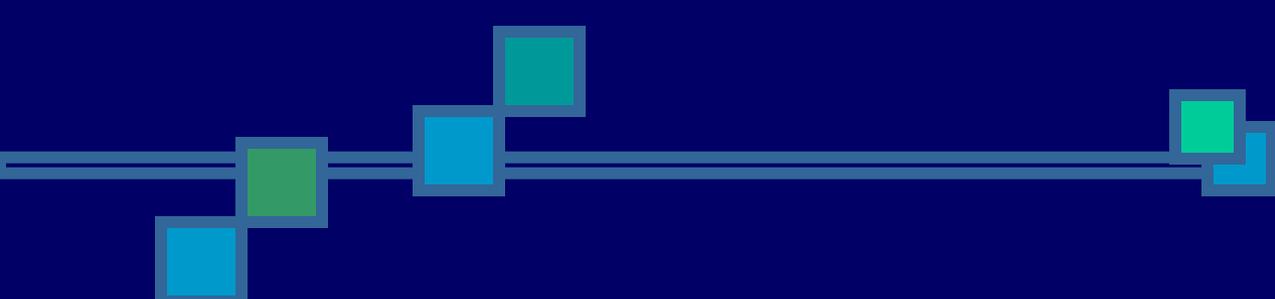


XML

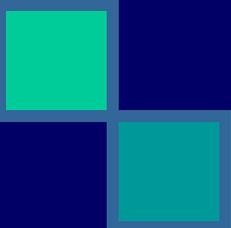


Grundbau Notenblatt

- Titel des Stückes
 - Name Komponist
 - Metronomangabe
 - Stimmenbezeichnung/Instrument
 - Notenzeile
 - Notenschlüssel
 - Tonart
 - Taktart
 - Takte
- 



XML



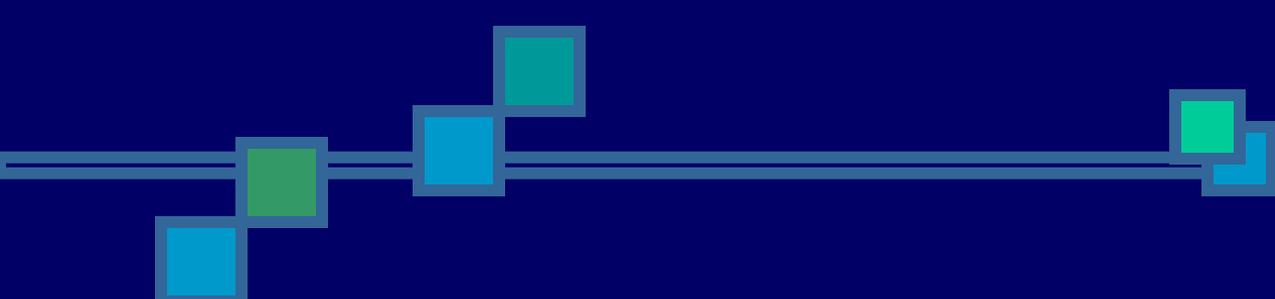
XML ist in der Tat gut für die Darstellung musikalischer Informationen der gezeigten Art geeignet, weil es die genannten musikalischen Konzepte darstellen und insbesondere die dazwischen bestehenden Zusammenhänge darstellen kann. Dazu zählen insbesondere die hierarchischen Beziehungen.

Hierarchische Beziehungen werden in der objektorientierten Sprechweise der Informatik als **Vererbungsbeziehungen** bezeichnet. In der Tat lassen sich solche Vererbungsbeziehungen schon an diesem Beispiel aufzählen:

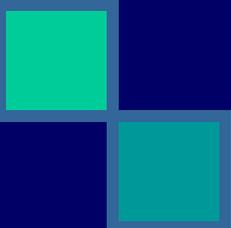


- der Titel gilt das ganze Stück
- der Schlüssel gilt für alle Zeilen und Takte
- die Tempobezeichnung gilt für alle Zeilen und Takte
- die Tonart gilt für alle Zeilen und Takte
- die Taktart gilt für alle Zeilen und Takte.

Man kann in objektorientierter Sprechweise beispielsweise sagen: die Takte des obigen Beispiels erben Schlüssel, Tonart und Taktart.

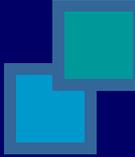


XML



Neben den hierarchischen Beziehungen gibt es in der Musik und in der üblichen Musiknotation noch die besonders wichtige linear - zeitliche Beziehung: Noten und Akkorde folgen ein-ander zeitlich mit definierten Abständen. Sie sind zeitlich geordnet. Üblicherweise stellt man diese Ordnung durch eine graphische Links-Rechts Beziehung dar. In XML kann man sowohl ungeordnete Mengen (ein Knoten kann beliebig viele Unterknoten haben, deren Reihenfolge nicht zwingend festgelegt ist) als auch Folgen (geordnete Mengen) wiedergeben, wobei natürlich nur dieser Aspekt (Notenereignisse sind linear geordnet) für die Musikwiedergabe wesentlich ist.

Neben der Wiedergabe von hierarchischen und linearen Beziehungen gibt es eine dritte Eigenschaft, die XML für die Wiedergabe von Musikereignissen besonders geeignet erscheinen lässt: XML-Knoten können Attribute enthalten, deren Zahl und Typ frei definierbar ist. Für einen Knoten des (frei definierbaren) Typs "Note" lassen sich z.B. die folgenden Attribute definieren:

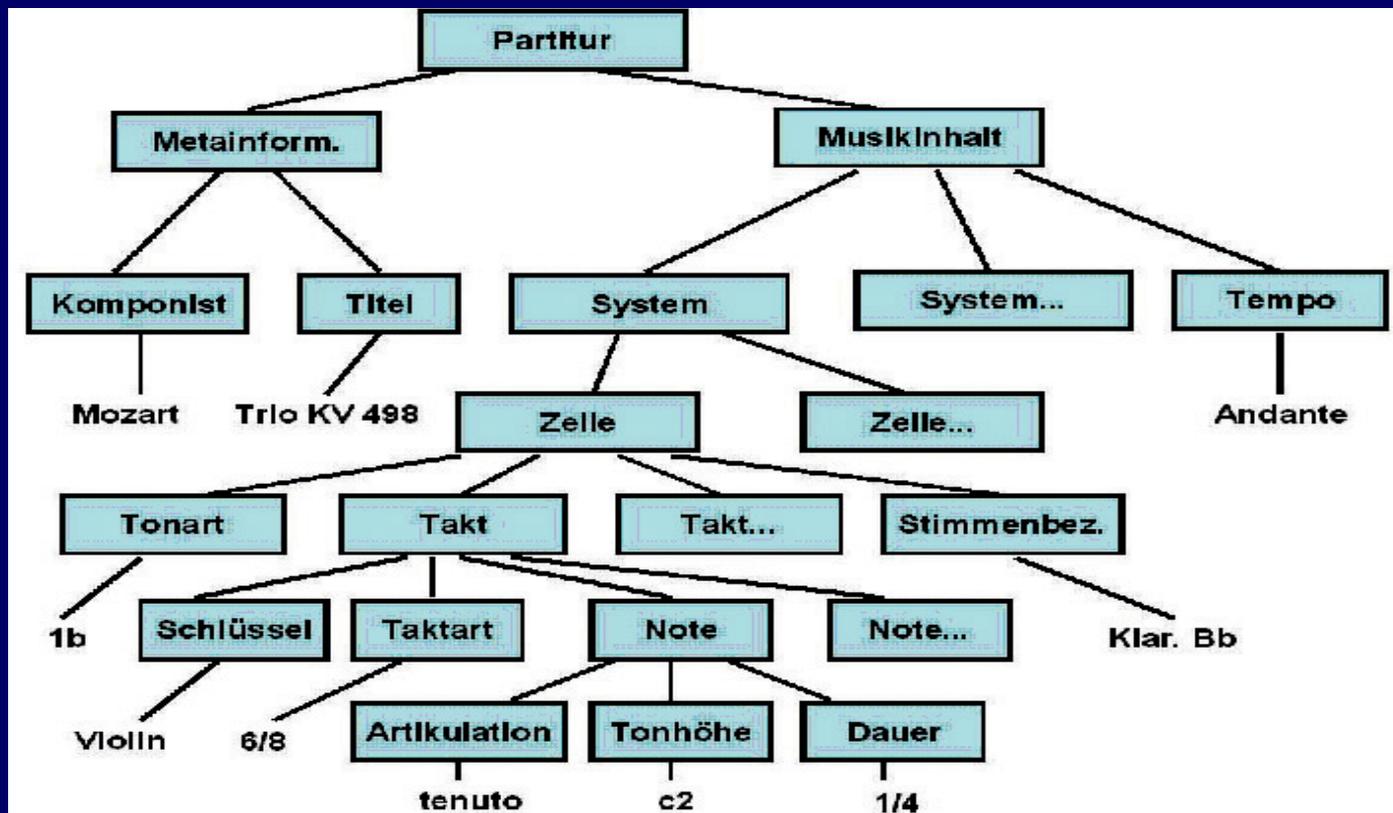


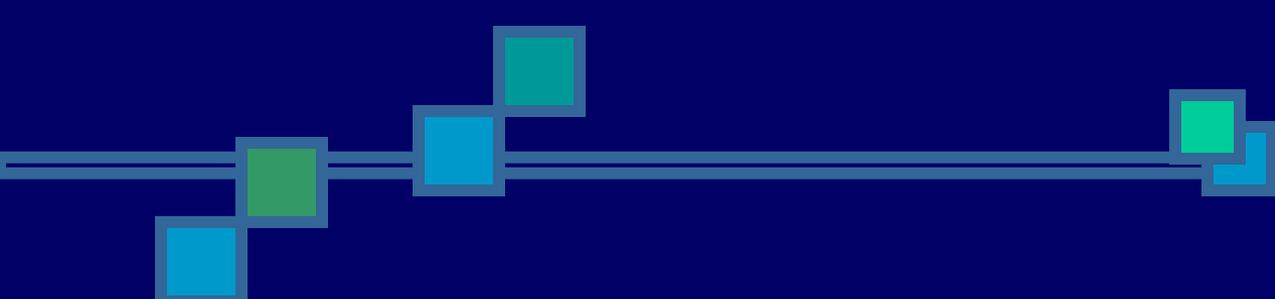
- Tonhöhe
- Name
- Dauer (z.B. absolut Millisekunden oder relativ in 64tel)
- Vortragsart (legato, staccato, akzentuiert).

In XML kann man dies z.B. so ausdrücken

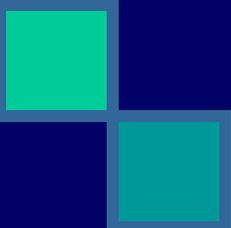
TON name="c" hoehe="2" dauer="16/64" vortrag="staccato"

XML



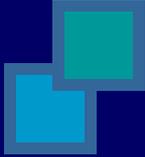


XML

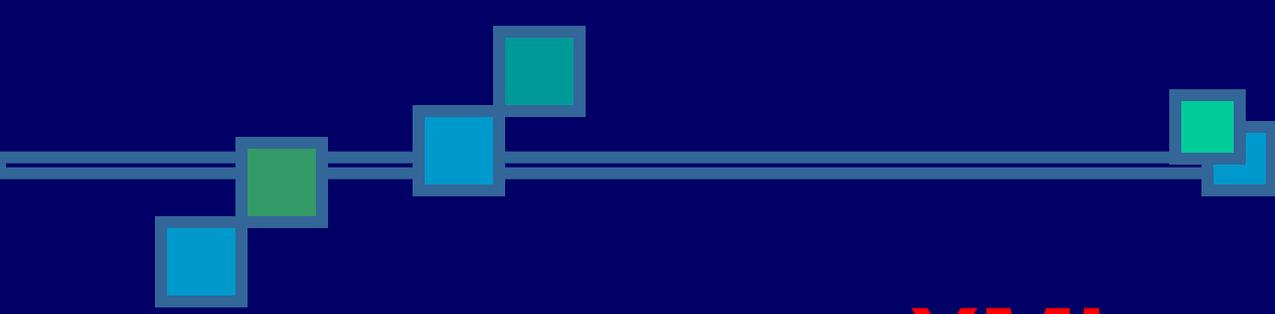


1. das XML-Dokument, das die Dokumentdaten enthält. In unserem Fall wären dies alle musikalischen Daten (die Noten), die Ausführungsvorschriften (Tempo etc.) und die Metainformationen (Komponist etc.), die im Notenbild erscheinen.

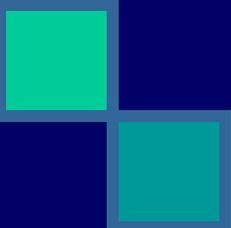
2. eine Syntaxdefinition (Grammatik) in Form eines XML-Schemas oder in Form einer document type definition (DTD). In dieser Datei sind Regeln festgelegt, wie XML-Elemente, Attribute und andere Daten definiert und mit einem logischen Bezug zueinander in einem XML-Dokument dargestellt werden können. Und noch wichtiger: anhand eines DTDs oder eines XML-Schemas kann ein Empfängerprogramm prüfen, ob das XML-Dokument wohlgeformt ist (syntaktisch ok) und somit gefahrlos verarbeitet werden kann. Name und Pfad des DTDs oder XML-Schemas werden übrigens im Dokumentkopf von XML-Dateien festgehalten.



3. Hinzu kommt sehr oft ein dritte Datei, eine sogenannte Stylesheet-Datei (z.B. im Format CSS, = "cascaded style sheet" aus dem WEB-Kontext). Sie legt fest, wie die Dokumentelemente am Ausgabegerät (Bildschirm, Drucker, Sequenzer etc.) dargestellt werden sollen. Diese Stylesheet-Datei kann für verschiedene Ausgabegeräte unterschiedliche Formatvorschriften enthalten, also etwa für Farbdrucker andere Formatierungsvorschriften wie für Schwarz-Weiß-Drucker. Stylesheet-Dateien sorgen also für die Trennung von Form und Inhalt, auf die man sich in der Welt der Textverarbeitung seit langem geeinigt hat.



XML



Ein Beispiel:

All the things you are (Jerome Kern)



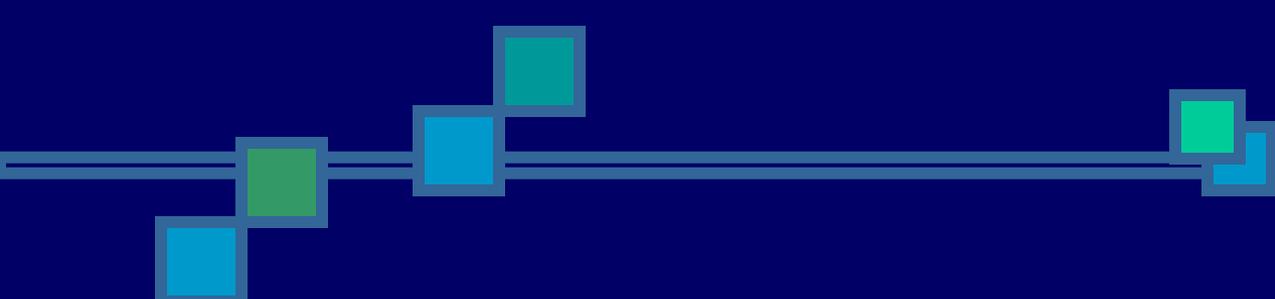
Analyse und Anwendung von XQUERY:
Vgl. K. Huckert: XML im Jazz (Aufsatz)

Stufenleiterakkorde in Dur

Tonart	I maj7	II m7	III m7	IV maj7	V 7	VI m7	VII m7b5
C-Dur	Cmaj7	Dm7	Em7	Fmaj7	G7	Am7	Bm7b5
F-Dur	Fmaj7	Gm7	Am7	Bbmaj7	C7	Dm7	Em7b5
Bb-Dur	Bbmaj7	Cm7	Dm7	Ebmaj7	F7	Gm7	Am7b5
Eb-Dur	Ebmaj7	Fm7	Gm7	Abmaj7	Bb7	Cm7	Dm7b5
Ab-Dur	Abmaj7	Bbm7	Cm7	Dbmaj7	Eb7	Fm7	Gm7b5
Db-Dur	Dbmaj7	Ebm7	Fm7	Gbmaj7	Ab7	Bbm7	Cm7b5
Gb-Dur	Gbmaj7	Abm7	Bbm7	Cbmaj7	Db7	Ebm7	Fm7b5
B-Dur	Bmaj7	C#m7	D#m7	Emaj7	F#7	G#m7	A#m7b5
E-Dur	Emaj7	F#m7	G#m7	Amaj7	B7	C#m7	D#m7b5
A-Dur	Amaj7	Bm7	C#m7	Dmaj7	E7	F#m7	G#m7b5
D-Dur	Dmaj7	Em7	F#m7	Gmaj7	A7	Bm7	C#m7b5
G-Dur	Gmaj7	Am7	Bm7	Cmaj7	D7	Em7	F#m7b5

Wichtige Akkordfolgen im Jazz

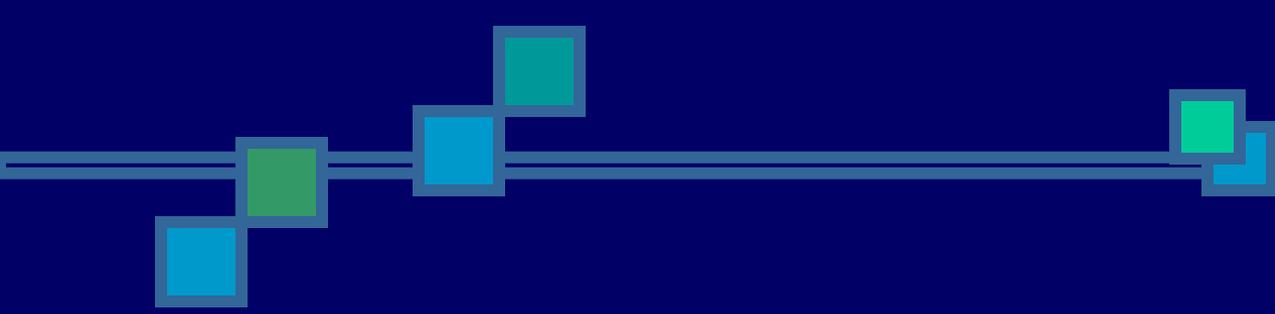
Verbindung	Kurzsprechweise
IIIm7 V7	(2-5)
V7 Imaj7	(5-1)
IIIm7 V7 Imaj7	(2-5-1)
IIIm7 V7 Imaj7 IVmaj7	(2-5-1-4)
VIm7 IIIm7 V7 Imaj7	(6-2-5-1)
IIIIm7 VIm7 IIIm7 V7 Imaj7	(3-6-2-5-1)
Imaj7 VIm7 IIIm7 V7	(1-6-2-5)



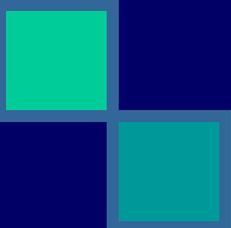
Vorteile computergestützter XML-Analyse

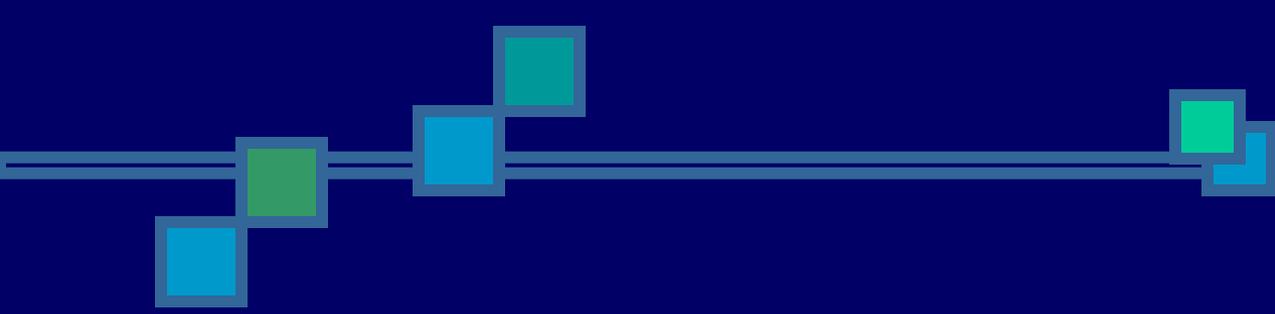
Durch die Analyse können sich Musiker auf die jeweilige Verbindung (Muster) einstellen und entsprechende Klischees zur Improvisation verwenden. Dies können geprobte Tonfolgen oder aber sogenannte Skalen (Kirchentonleitern) sein, die dann einen Teil der Improvisation bilden. Weiterhin können durch solche Muster Bewegungsabläufe – wie etwa das Greifen von Akkorden auf einer Gitarre – „automatisiert“ werden. Natürlich finden in der Improvisation weitere Techniken (z.B. Chromatik, Arpeggien etc) Verwendung, die hier nicht betrachtet werden.





XML und XQuery

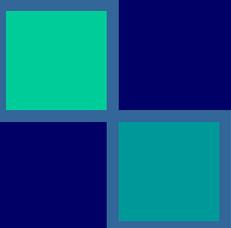
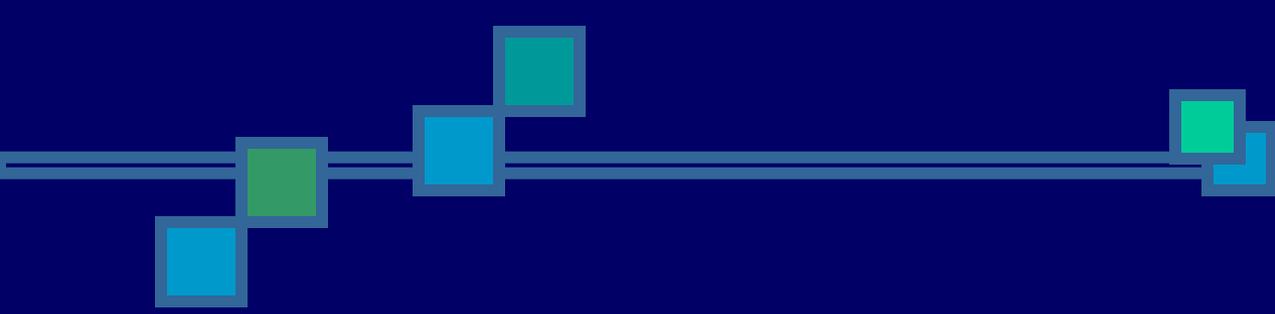
- 
1. Erkennung von typischen Akkordfolgen (2-5-1)
 2. Erkennung von Tritonussubstitutionen
 2. Algorithmus zur Erkennung eines Tongeschlechtswechsels
 3. Algorithmus zur Erkennung von Sekundär-Dominanten
 4. Algorithmus zur Erkennung von Akkord-Synonymen (siehe Sagmeister S. 184)
- 



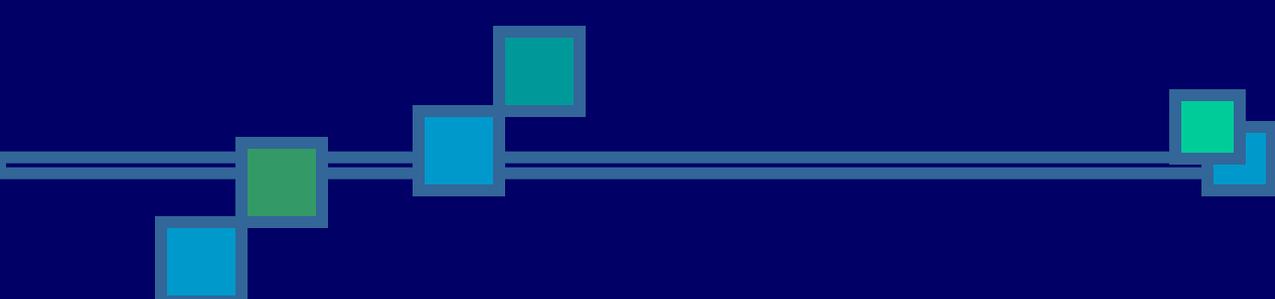
XML und XQuery

Livedemonstration mit dem SQL-
Server 2008

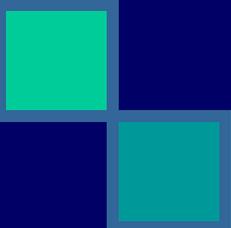
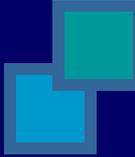
basierend auf 'All the things you are'



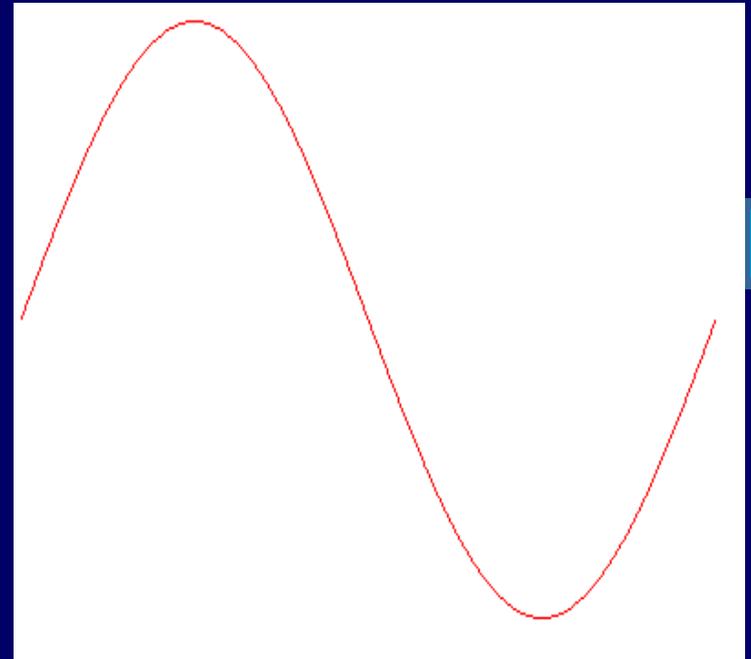
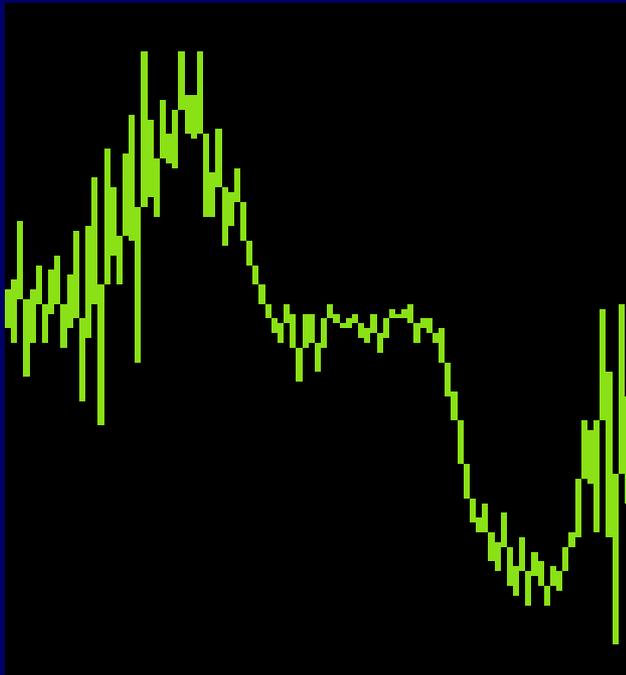
Audio, MP3 und Midi

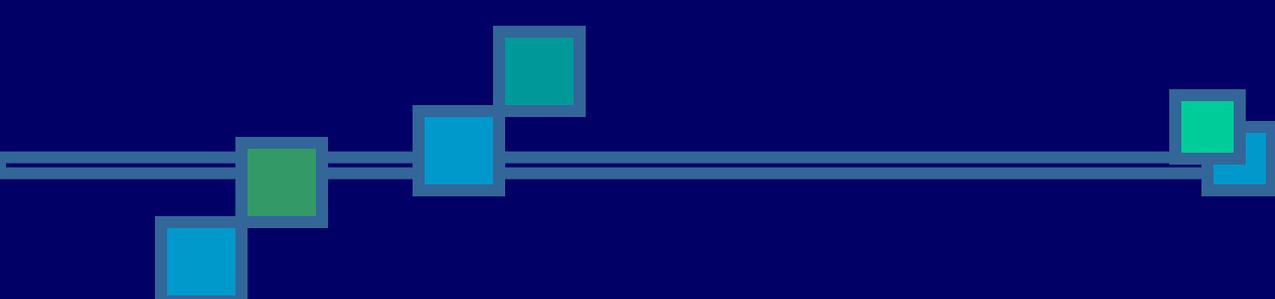


Grundlagen von Audio

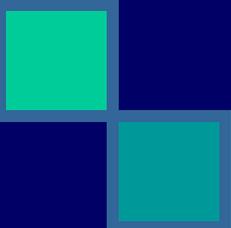
- 
- Audio = Geräusche, Sprache, Musik
 - Audio kann über Schallwellen definiert werden
 - Audio kann unter Zuhilfenahme von Audio-Hardware (Soundkarte, Mikrophon) im Computer gespeichert werden
- 

Audio Grundlagen (Obertonspektrum und Sinuston)

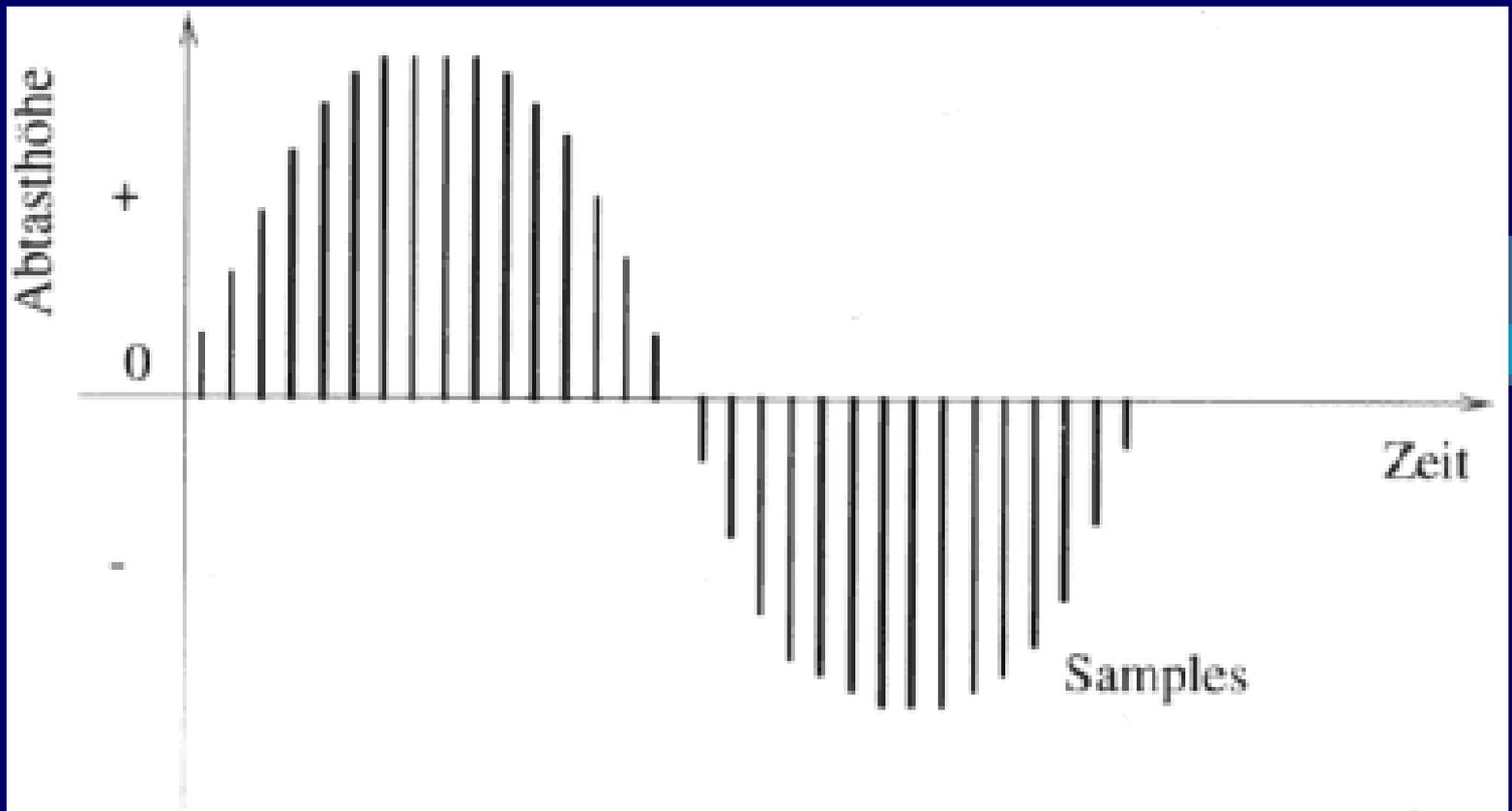


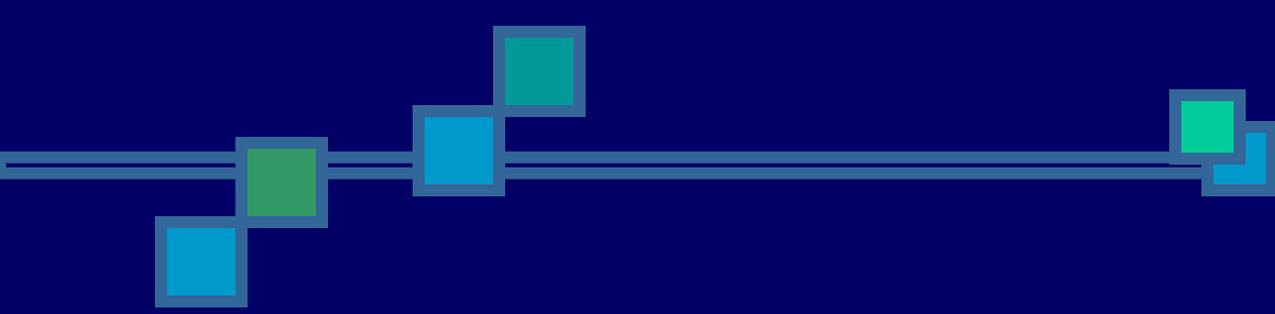


Grundlagen von Audio

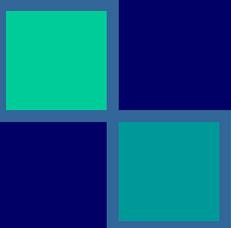
- 
- Digitalisierung
 - Umwandlung analoges Signal (kontinuierlich) in Folge von diskreten Werten
 - Analoges Signal muß im Frequenzbereich bandbegrenzt werden (für Menschen von 20 Hz bis ca. 20 kHz)
- 

Grundlagen von Audio





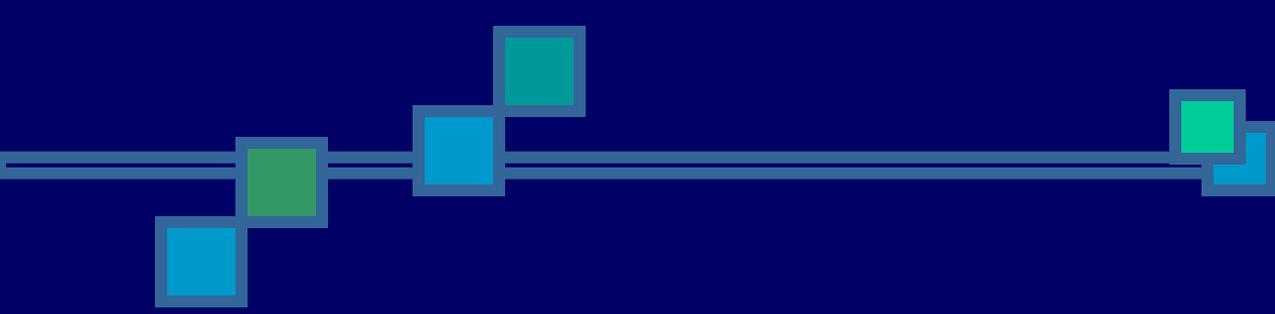
Grundlagen Audio



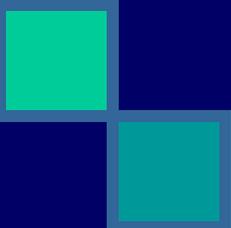
Abtasttheorem von Shannon/Nyquist:

besagt, daß ein kontinuierliches Signal mit diskreten Werten dargestellt werden kann, solange es nur bandbegrenzt ist.





Grundlagen Audio (Begriffe)



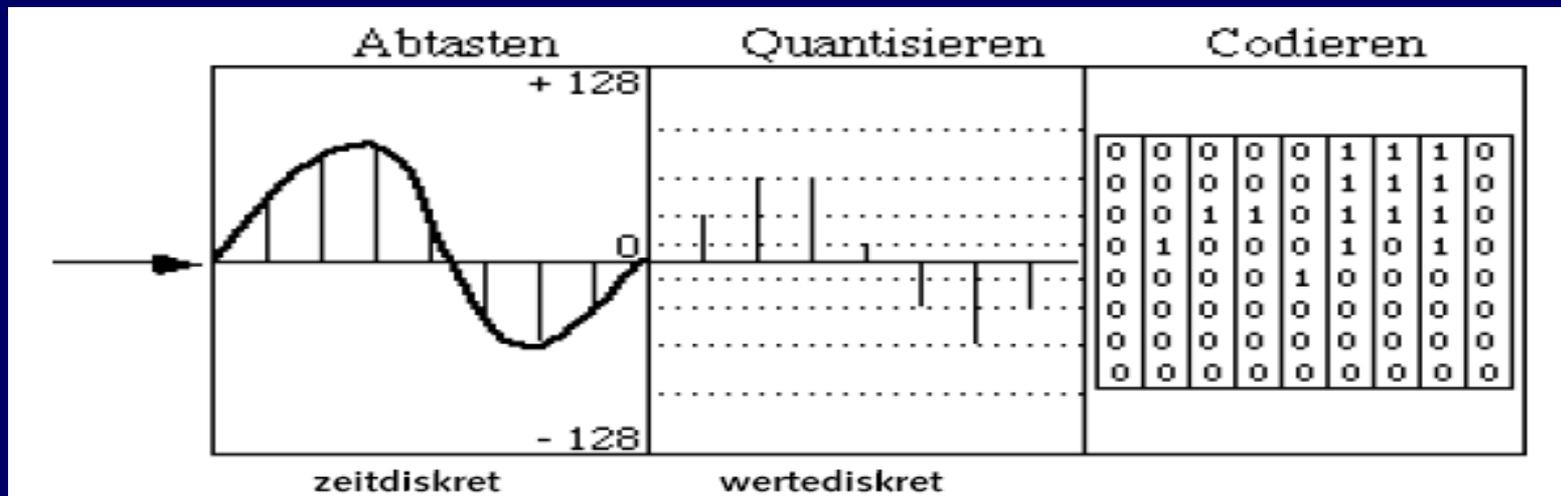
- Sampling (Abtastung)

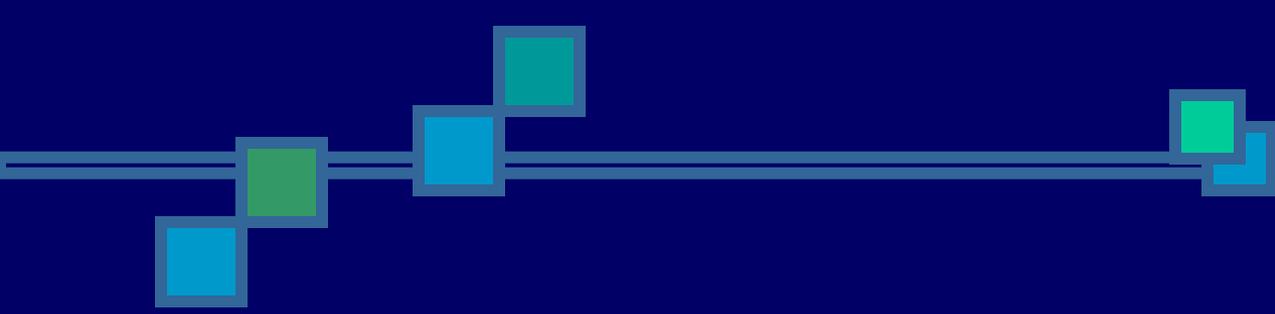
Abtastung (englisch *sampling*) ist die Registrierung von Messwerten zu diskreten, meist äquidistanten Zeitpunkten. Aus einem zeitkontinuierlichen Signal wird so ein zeitdiskretes Signal gewonnen. Die Anzahl der Abtastungen pro Sekunde wird Abtastrate (Samplingrate) genannt.



Grundlagen Audio

- Jeder Sample wird mit einem Binärwort codiert (PCM Pulse Code Modulation)



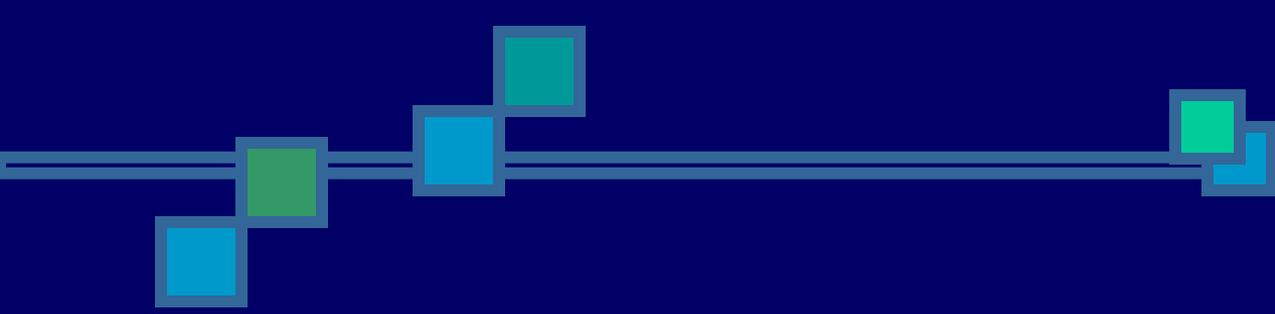


Grundlagen Audio (Begriffe)

- Beispiel

Bei CD-Qualität beträgt die Abtastrate 44100 Hz, d.h. das Signal wird 44100-mal abgetastet. Diese Rate folgt aus dem Abtasttheorem, das besagt, daß die Samplerrate mindestens doppelt so hoch sein muss wie die höchste im Signal enthaltene Frequenz. (Mensch 20.000 Hz)



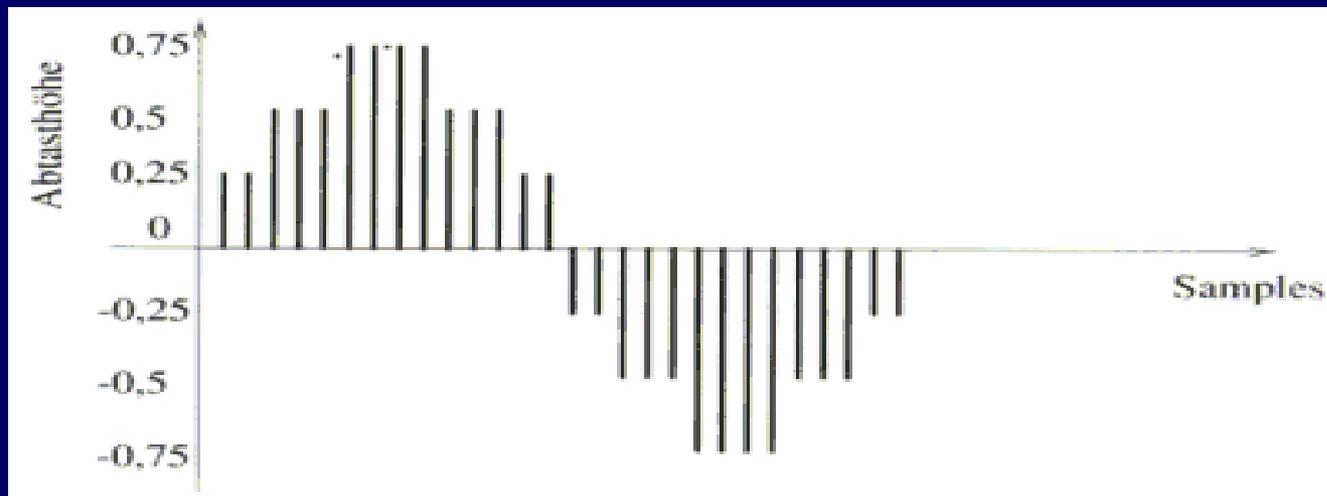


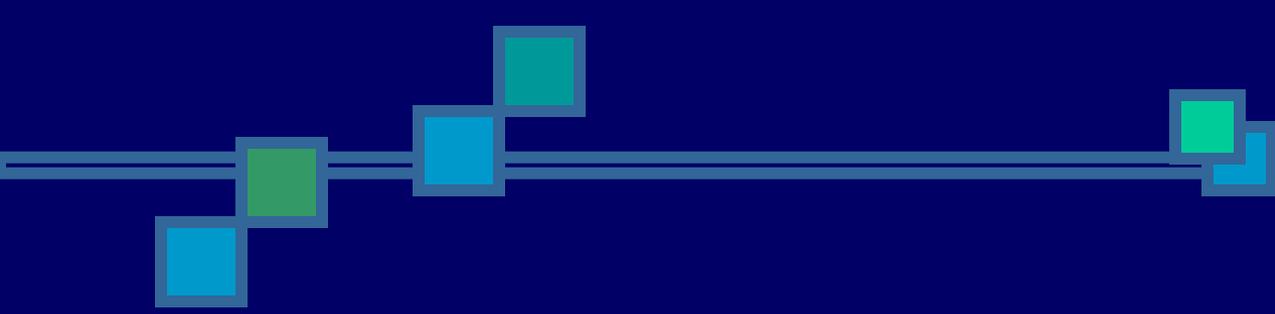
Grundlagen Audio (Begriffe)

- Wortbreite bei Audio (Samplewort) zwischen 8 und 24 Bit
 - Rasterung des kontinuierlichen Audiosignals in eine endliche Anzahl von digitalen Zahlenwerten wird **Quantisierung** genannt.
- 

Grundlagen Audio (Begriffe)

3 Bit-Quantisierung



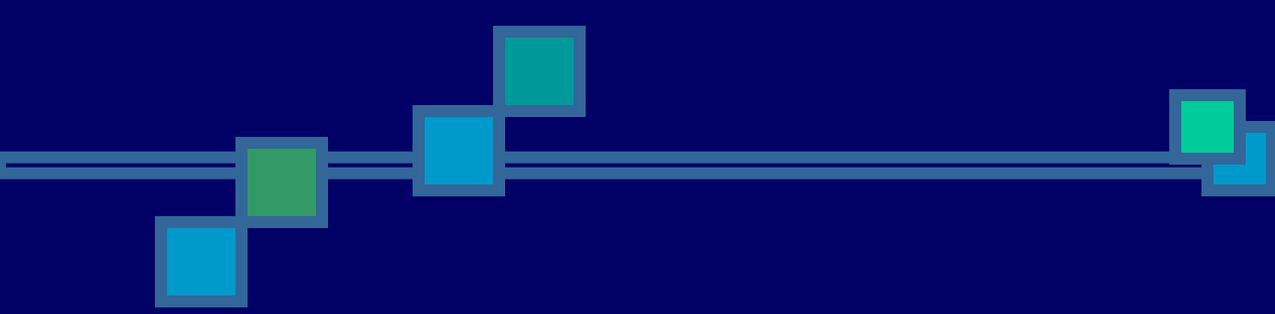


Grundlagen Audio (Begriffe)

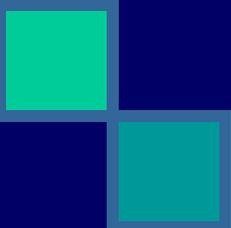
- Quantisierungsfehler

Bei der Abtastung des Audiosignales wird im vorhinein festgelegt, wie viele verschiedenen Werte man für die Höhe der Amplitudenwelle abspeichert. Diese Quantisierung legt also fest, wie stark man die festgestellten Werte runden muss. (Quantisierungsfehler). Macht sich als Rauschen bemerkbar.





Grundlagen Audio (Begriffe)



Datenmengen bei Speicherung

(Auflösung (Bit)) / 8 Bit * Kanäle * Sampling-Frequenz

Beispiel:

CD-typische Auflösung von 16 Bit, 2 Kanäle Stereo, Sampling-frequenz von 44,1 kHz

$2 * 2 * 44100 = 176$ Kbytes pro Sekunde

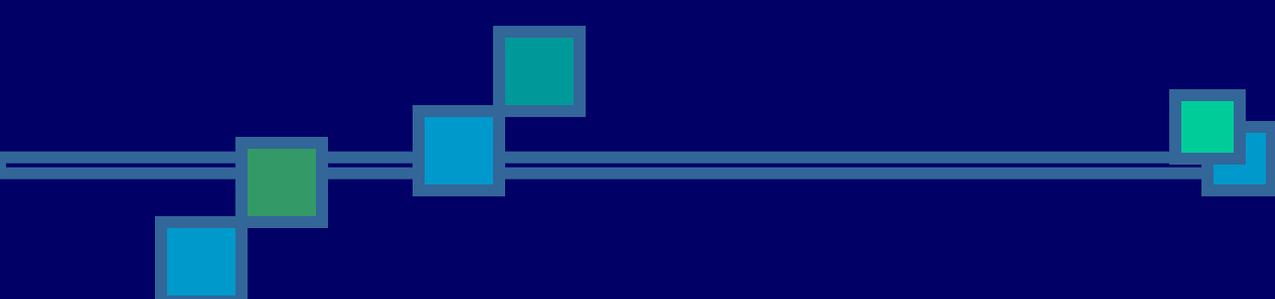
Grundlagen Audio (Begriffe)

	11 kHz	22 kHz	44,1 kHz
8 Bit	22 KB/s	44 KB/s	88 KB/s
16 Bit	44 KB/s	88 KB/s	176 KB/s

Grundlagen Audio (Audiokompression)

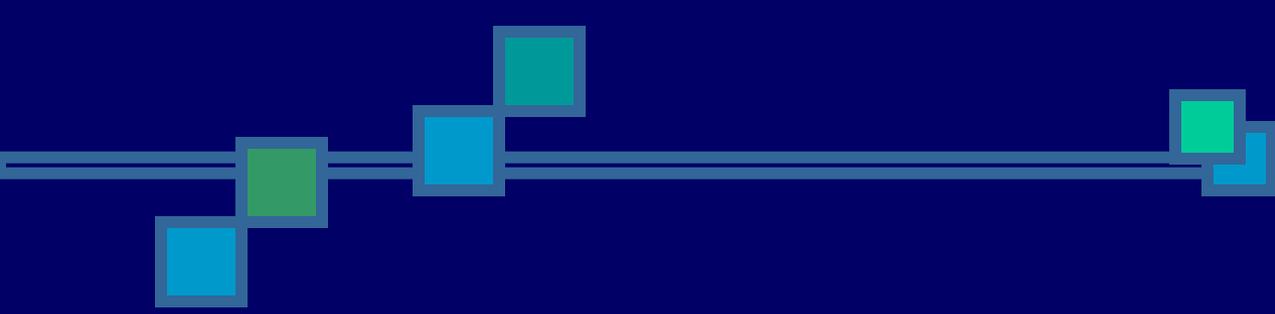
- Der französische Mathematiker Joseph Fourier (1768-1830) entwickelte ein mathematisches Verfahren, um Signale aus der Zeit- in die Frequenzdarstellung zu transformieren – die Fouriertransformation.
- Ein Algorithmus, der diese Transformation sehr schnell im Computer ausführt, ist die Fast Fourier Transformation (FFT) durch Anwendung des Prinzips „Teile und Herrsche“.
- Sie ist ein wichtiges Hilfsmittel für die Audiokompression, da die wesentlichen Kompressionsstufen auf der Frequenzdarstellung basieren.





Grundlagen Audio

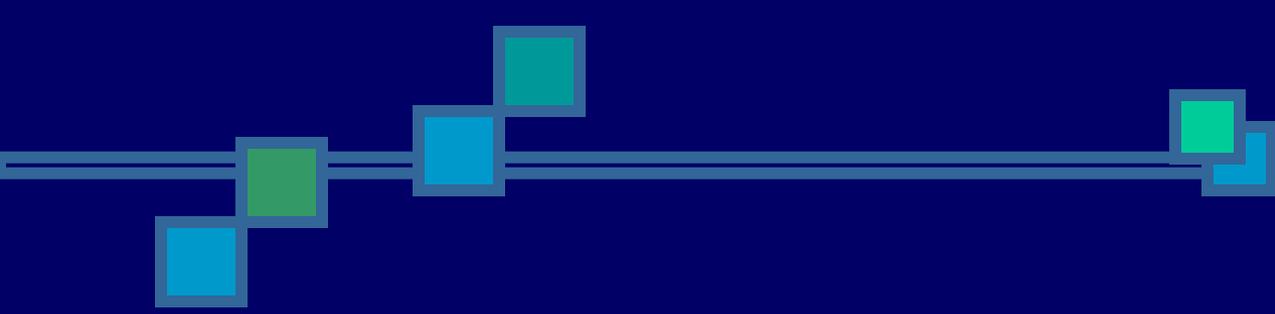
- Arbeiten mit digitalisierten Audio-Daten
 - Speichern der Daten
 - Editieren (neu zusammenfügen, Änderung Klang)
 - Time Stretching
 - Tonhöhenkorrektur
 - Datenkompression
 - Entrauschen und Restaurieren
 - Klangverschmelzungen
- 



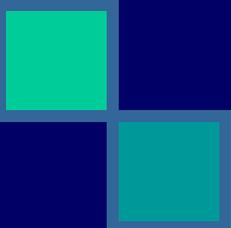
Audio, MP3 und MIDI



Codierung digitaler Audiodaten

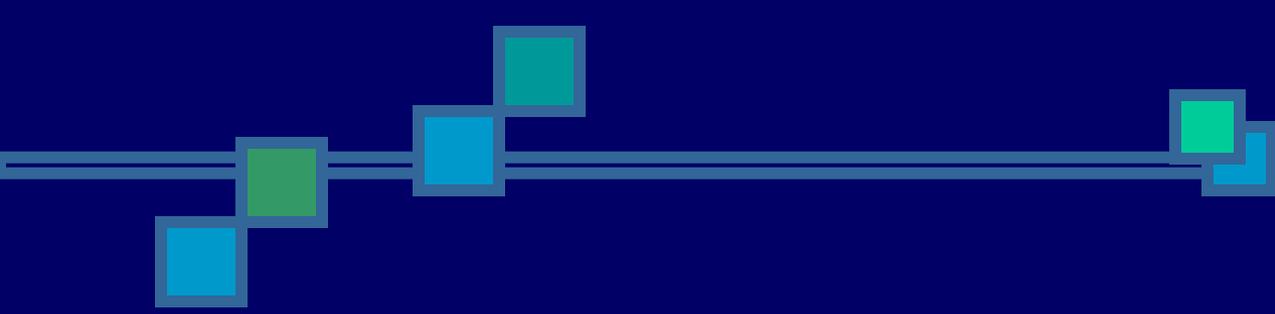


Digitalformate



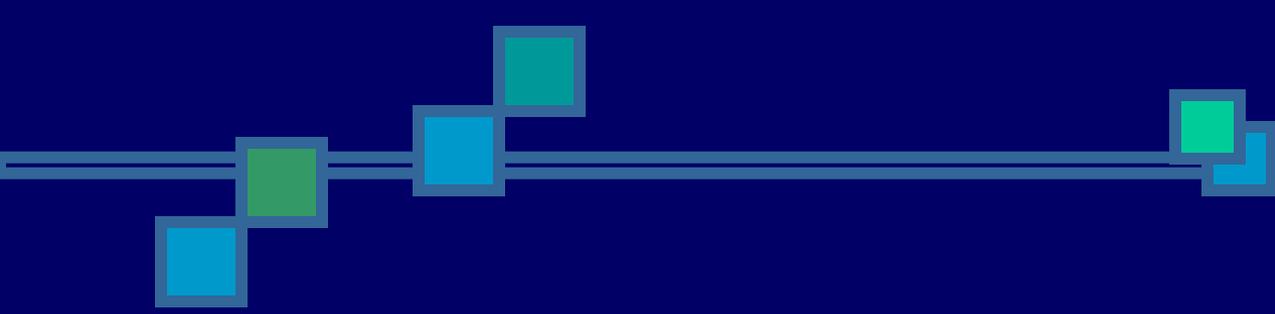
- Audio-Dateiformate

- WAV-Format zum Speichern von Audiodaten auf Platte, IBM/Microsoft
 - AIFF (Audio Interchange File Format), Apple
 - SDII (SoundDesigner II)
 - MPEG (z.B. MP3)
 -
- 

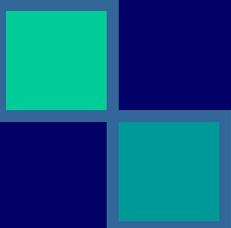


Audio (WAV-Format)

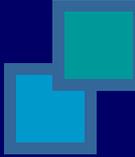
- Standard-Format für Audiodaten unter Windows
 - Herkunft aus EA IFF85 Format
 - Besteht aus CHUNKS (Häppchen, Segmenten)
- 

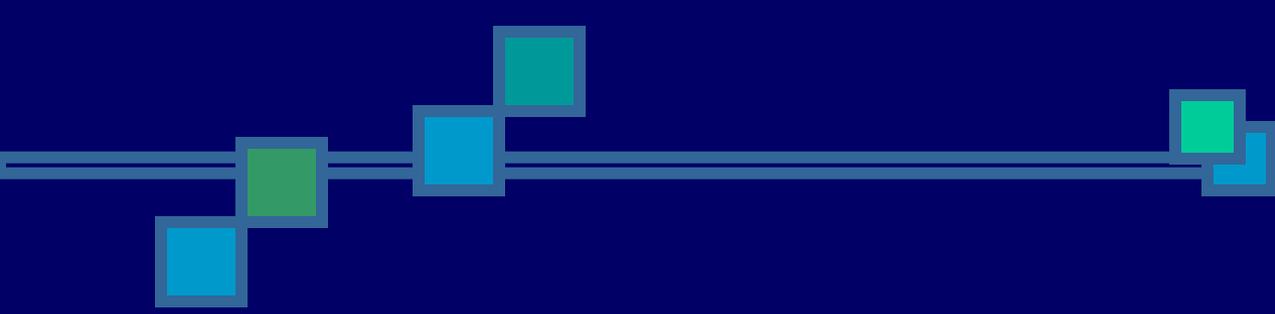


Audio (WAV-Format)

- 
- Wichtige Internetquelle

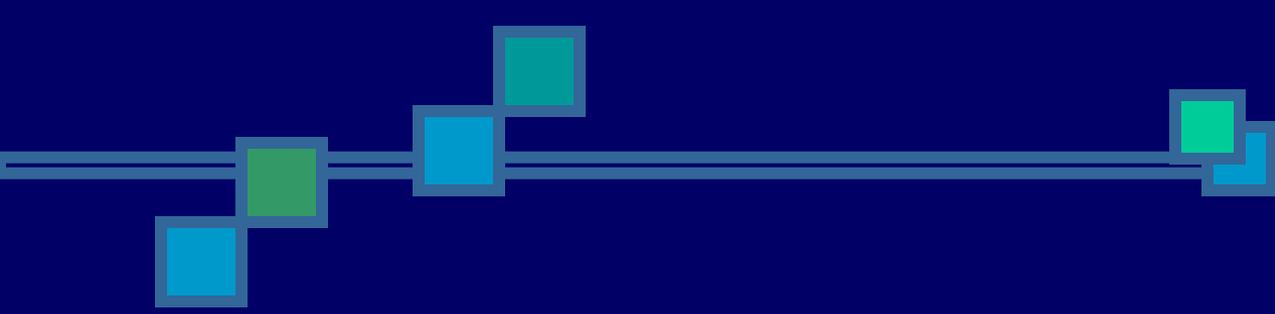
<http://www.lightlink.com/tjweber/StripWav/WAVE.html>



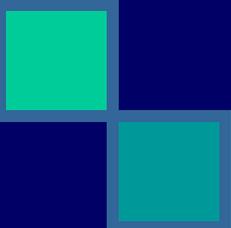


Audio (WAV-Format)

- Chunk besteht aus
 - 4 Byte Identifikation
 - Angabe der Größe der Datei (4 Bytes)
 - Chunk-Daten
- 

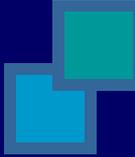


Audio (WAV-Format)



Genereller Aufbau

RIFF-Header
FORMAT-Chunk
DATA-Chunk



Audio (WAV-Format)

Aufbau RIFF-Chunk

Offset	Bytes	Anmerkungen
00h	4	Chunk-Name RIFF
04h	4	Chunk-Länge
08h	4	RIFF-Typ WAVE 59

Audio (WAV-Format)

Aufbau FORMAT-Chunk

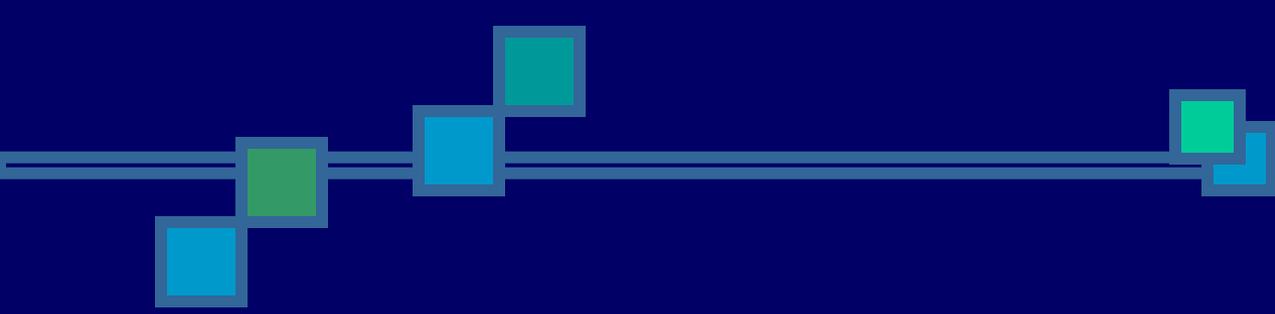
Offset	Bytes	Anmerkungen
00h	4	Chunk-Name FMT
04h	4	Chunk-Länge
08h	2	0 = mono, 1 = stereo
0Ah	2	Kanalzahl
0Ch	4	Sample-Rate
10h	4	Byte pro Sekunde
14h	2	1 = 8 bit mono, 2 = 8 Bit stereo, 4 = 16 Bit stereo
16 h	2	Bits pro Sample

60

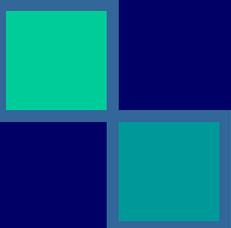
Audio (WAV-Format)

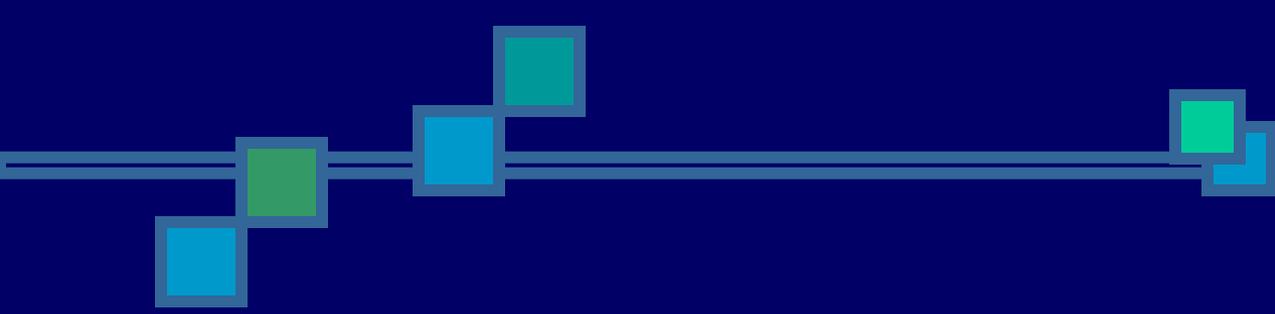
Aufbau DATA-Chunk

Offset	Bytes	Anmerkungen
00h	4	Chunk-Name DATA
04h	4	Chunk-Länge
08h	N	Datenbereich

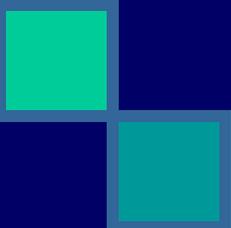


Audio (WAV-Format)

- 
- Bemerkungen zu WAV
 - Vielzahl von CHUNK-Typen
 - Möglichkeit der komprimierten Speicherung (μ -Law, DPCM, ADPCM)
 - In der Regel können Daten um Faktor 2 komprimiert werden
- 



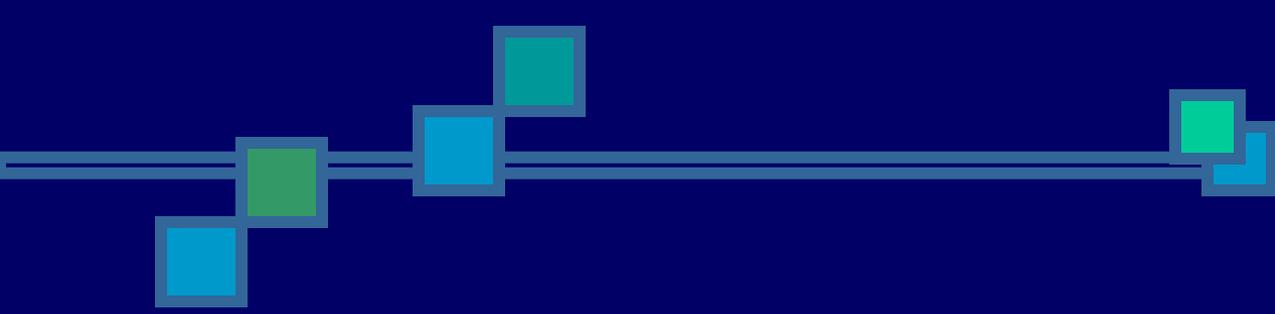
Audio (WAV-Format)



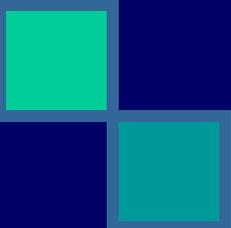
- Praktikum

Untersuchen Sie den WAV-File WIND.WAV mit einem Hex-Editor und interpretieren Sie das Ergebnis



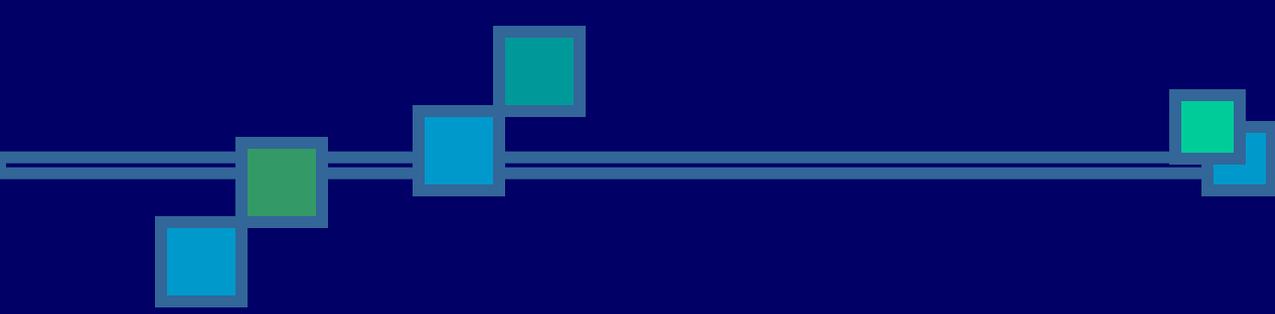


Grundlagen Audio



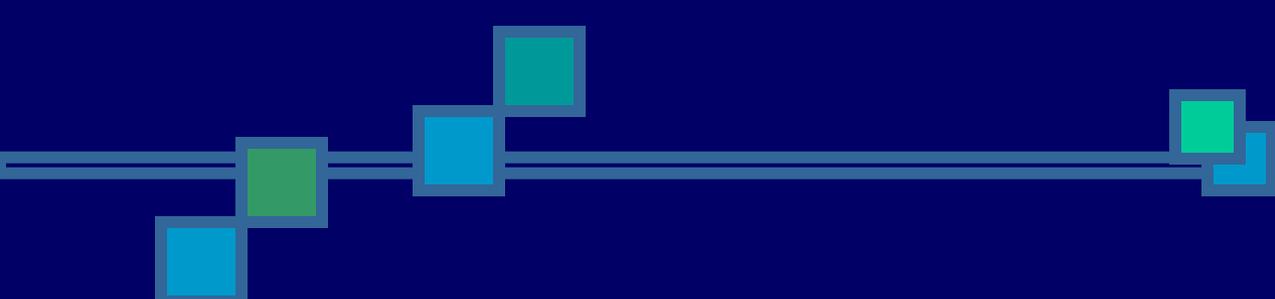
Fallstudie Softwarepaket Accoustica
(www.aconas.de)





Praktikum Audio

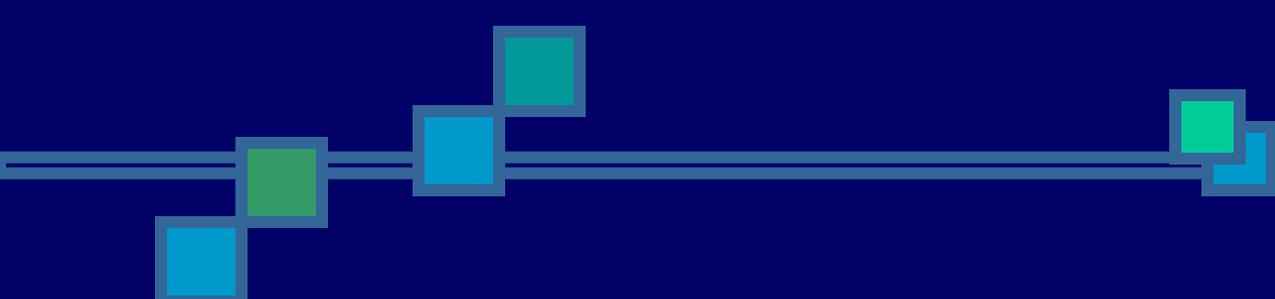
- Acoustica
 - Aufnahme von Audiodaten
 - Hinzufügen von Effekten (Echo, Chorus, Flanger, Modulator, Verzerrung, Limiter)
 - Restauration (Übertragung von LPs und Bändern)
 - Entfernen von Rauschgeräuschen
 - Transponierung, Harmonisierung, Timestretching



Praktikum Audio

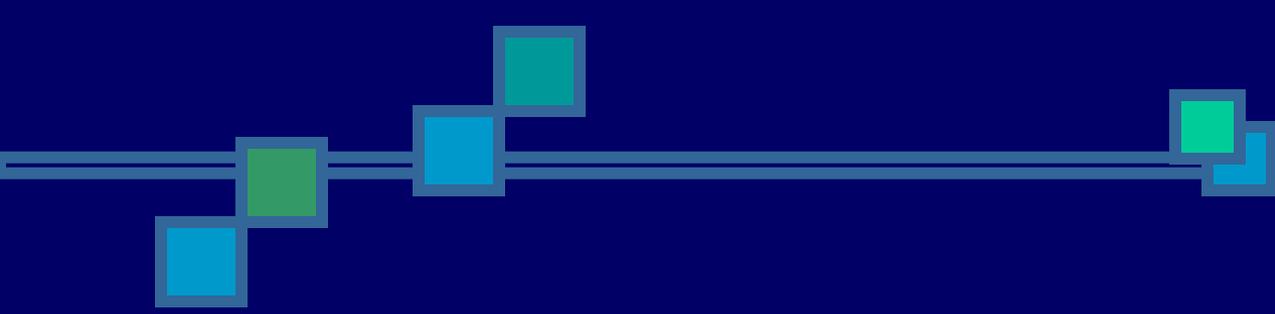
■ Wichtige Begriffe

- **Limitier:** begrenzen Signalstärke, um Verzerrungen zu umgehen
- **Echo:** Originalsignal und Reflexion an einem Gegenstand (z.B. Berg, Hauswand)
- **Hall:** Originalsignal wird mehreren Gegenständen in verschiedenen Abständen reflektiert
- **Chorus:** Effekt, der einen Ton so ausprägt, als würde gleichzeitig ein zweiter ähnlicher Ton mitklingen und sich dabei im Raum bewegen. Chorus-Effekt entsteht durch zeitversetzte Kopie Originalton, der aber gegenüber Originaltonhöhe etwas versetzt ist. Erzeugt warmen und vollen Klangeindruck.

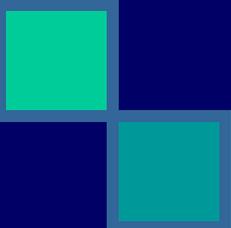


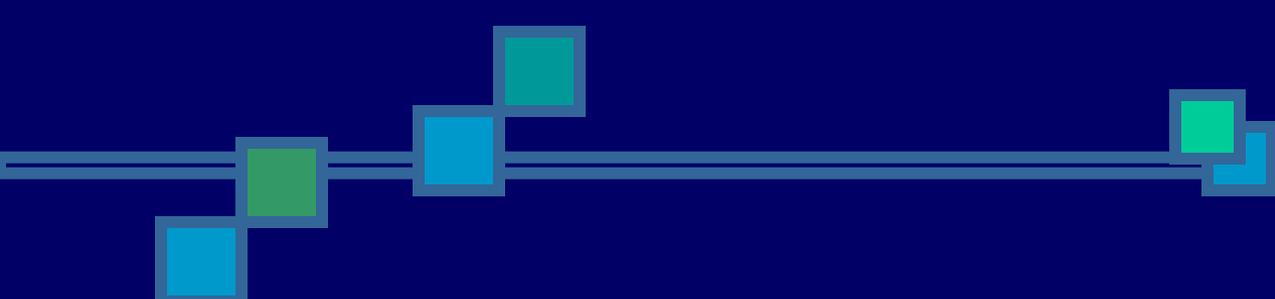
Praktikum Audio

- **Wichtige Begriffe:**
- **Flanger:** ähnlich Chorus, legt aber mehr wert auf Klangverfärbungen, schwebender Charakter
- **Modulator:** Chorus + Flanger + Phaser
- **Harmonisierer:** legt bis zu vier transponierte Stimmen übereinander
- **Transposition:** verändert Tonlage/Tonart um eine gewisse Höhe (Sekunde, Terz,, Oktav)
- **Timestretching:** verändert Länge einer Aufnahme (z.B. durch Tempoänderung)
- **Umkehrung:** Audio-Datei wird rückwärts abgespielt

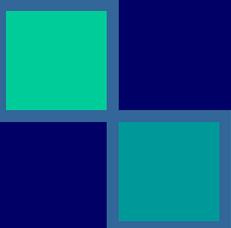


Praktikum Audio

- 
1. Laden Sie die Datei „Sunny“ im Verzeichnis Musik und Computer
 2. Spielen Sie diese ab
 3. Testen Sie die unterschiedlichen Effekte
 4. Machen Sie diesen Effekt rückgängig
 5. Ähnliche Aktionen mit S` Wonderful und Gentle Rain
- 



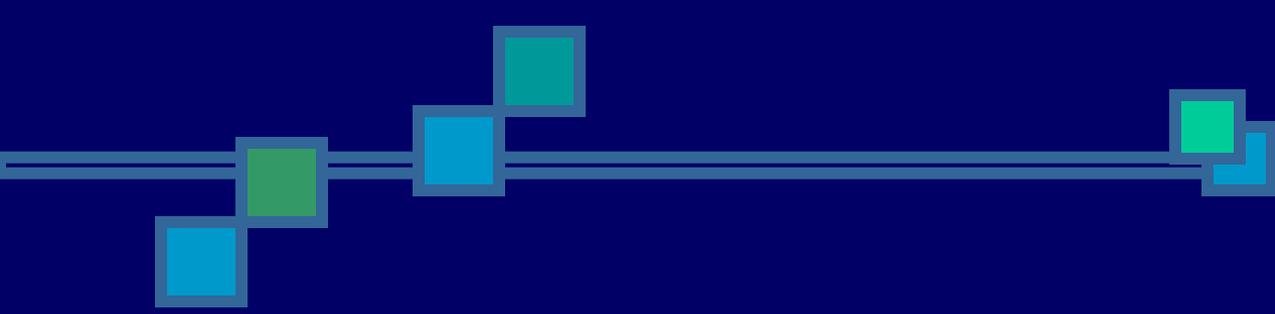
Praktikum Audio

- 
- 6. Ändern Sie die Lautstärke des gesamten Stückes
 - 6. Faden Sie die Aufnahme linear und exponentiell
 - 7. Laden Sie den Originaldateien und transponieren Sie diese um eine Terz
 - 8. Time-Stretching und Anwendungen
- 

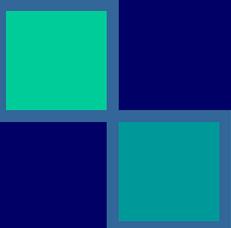
MP3

70

23.08.2015



Audio (MP3)



Folgende Unterlagen wurden benutzt:

Robert Plato: Mathematik in MP3-Playern.

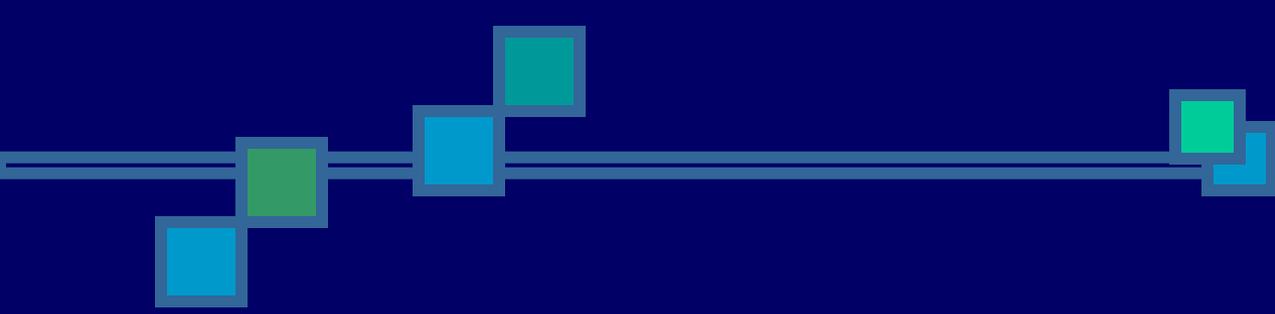
(Internet: www.mkath.uni-siegen/~plato/stt.pdf)

Nico Hartmann: MP3-Grundlagen: Psychoakustik.

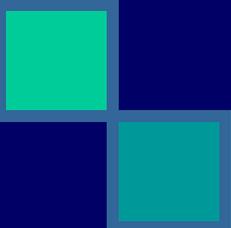
(Internet: www.tecChannel.de/multimedia/58/index.html)

Malte Jeschke: MP3-Grundlagen: Aufbau und Funktion. (Internet:

www.tecChannel.de/57/index.html)



Audio (MP3)



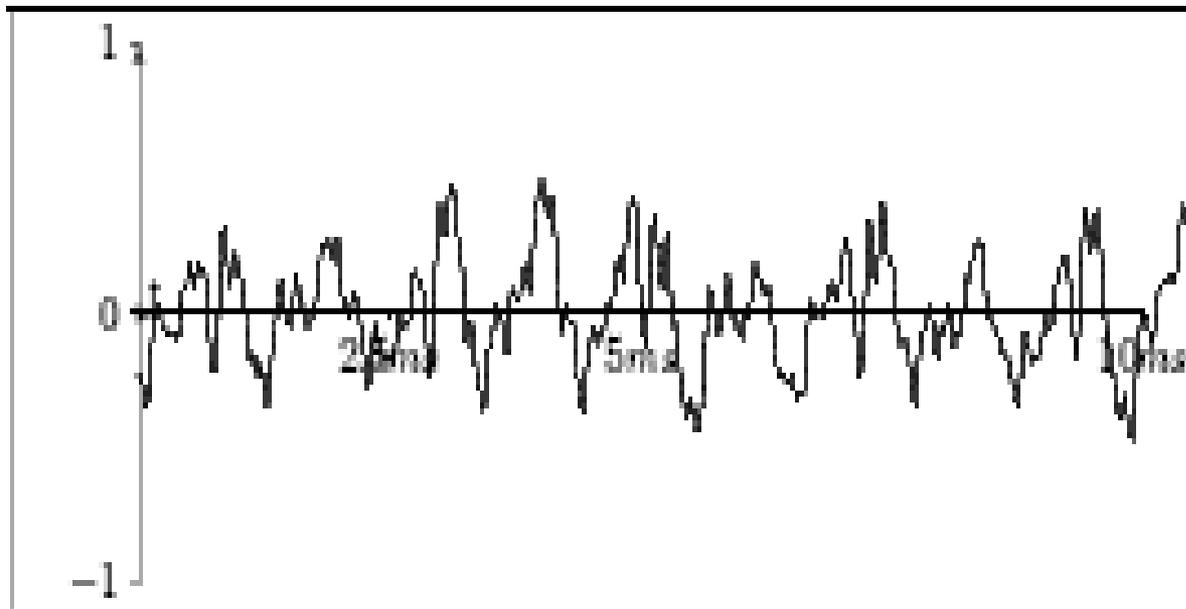
Ein Audiosignal erzeugt Schallwellen. Der von der Schallwelle verursachte Schalldruck wird in **Dezibel**, kurz dB, gemessen und nimmt typischerweise Werte zwischen 0 Dezibel (Stille) und 120 Dezibel (Schmerzgrenze) an. Eine Erhöhung des Schalldrucks um 10 Dezibel wird als eine Verdopplung der Lautstärke wahrgenommen.

Der Schalldruck lässt sich grafisch in Form eines Kurvenverlaufs $f(t)$ darstellen. Dieser Kurvenverlauf wird üblicherweise skaliert dargestellt mit Werten zwischen -1 und 1. Dabei erhalten

Werte erhöhten Drucks ein positives Vorzeichen, die Werte verminderten Drucks ein negatives Vorzeichen

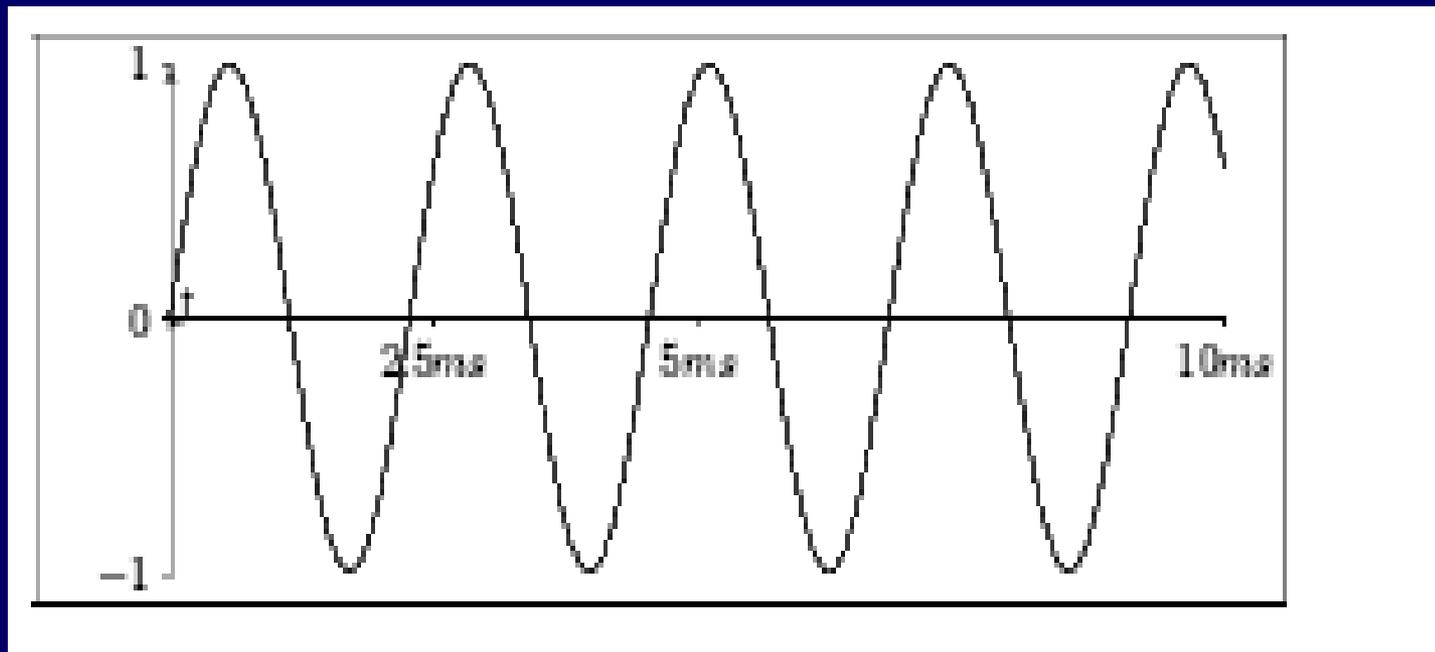
Audio (MP3)

Beispiel Shania Twain: Forever and for always



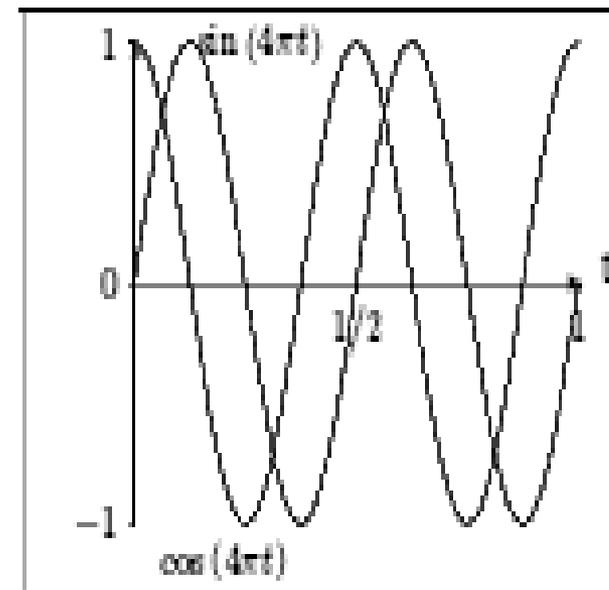
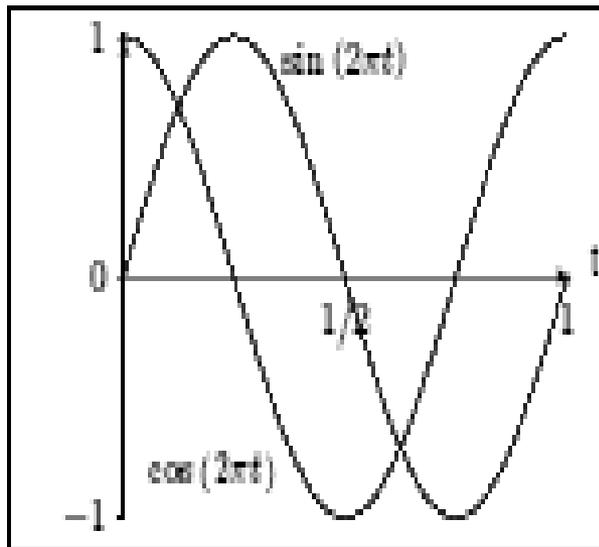
Audio (MP3)

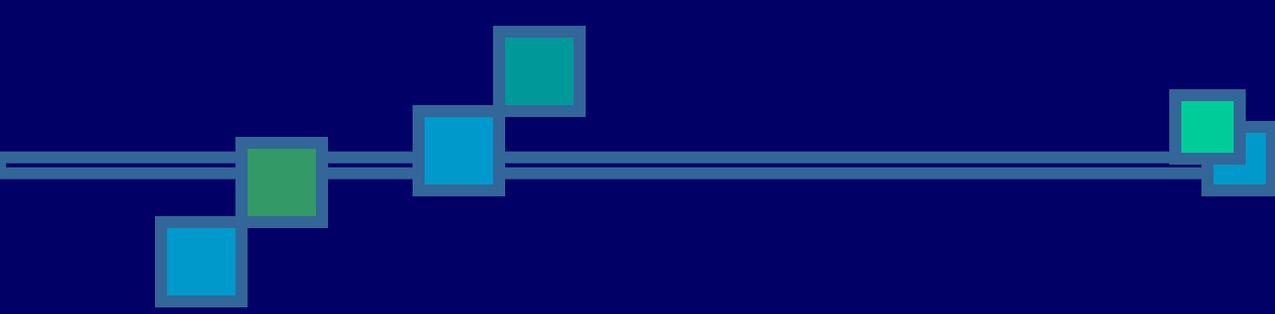
Kammerton A (Sinusschwingung 440 Hertz)



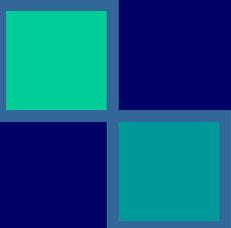
Audio (MP3)

Sinus-/Cosinusfunktion in Abhängigkeit von t





Audio (MP3)



Betrachte allgemeines Audiosignal $f(t)$ für
 $0 \leq t \leq 1$

Audiosignal kann interpretiert werden als
Überlagerung von viele gewissen Sinus-
und Cosinus-Funktionen.

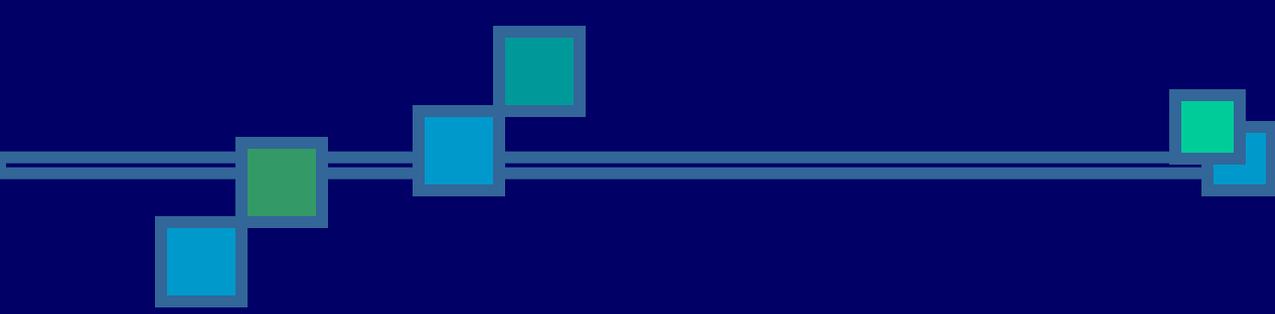


Audio (MP3)

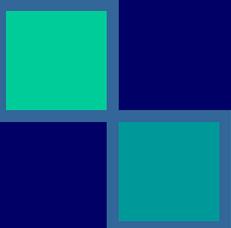
$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{22050} (a_k \cdot \cos(k \cdot 2\pi \cdot t) + b_k \cdot \sin(k \cdot 2\pi \cdot t))$$

$$a_k = 2 \cdot \int_0^1 f(t) \cdot \cos(k \cdot 2\pi \cdot t) dt \quad \text{und}$$

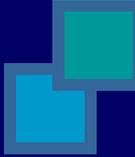
$$b_k = 2 \cdot \int_0^1 f(t) \cdot \sin(k \cdot 2\pi \cdot t) dt$$

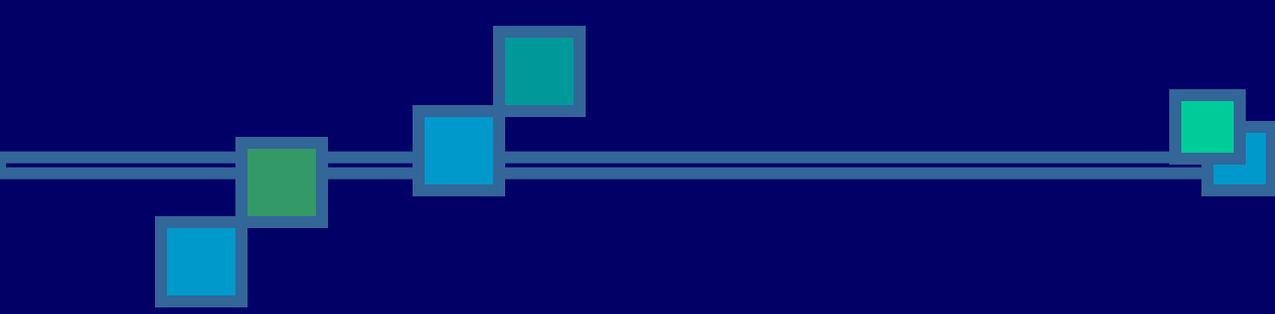


Audio (MP3)

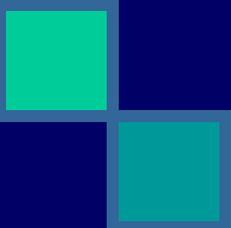


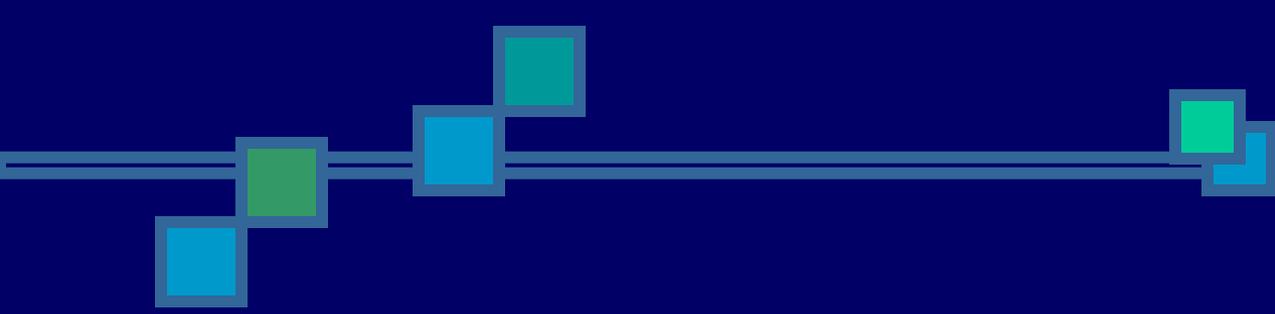
Die Bestandteile der einzelnen Summanden haben dabei die folgenden Bedeutung:

- (a) Die Zahl k ist die **Frequenz** der zugehörigen Sinusschwingung $\sin(k2\pi t)$ beziehungsweise der Cosinusschwingung $\cos(k2\pi t)$. Diese beiden genannten Schwingungsverläufe wiederholen sich in dem betrachteten Zeitraum $0 \leq t \leq 1$ jeweils k -mal.
 - (b) Die Koeffizienten a_k und b_k stellen die Amplituden dieser der beiden genannten Schwingungen dar. Je größer diese Koeffizienten betragsmäßig ausfallen, umso stärker sind die Frequenzen in dem Audiosignal in dem betrachteten Zeitintervall $0 \leq t \leq 1$ vertreten.
- 

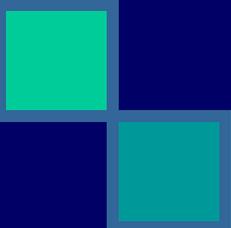


Audio (MP3)

- 
- Das menschliche Gehör nimmt Frequenzen etwa von 20 Hz bis 20 KHz wahr, insofern ist die Darstellung (1) plausibel.
 - Die Summanden zu den großen Werten von k machen die hochfrequenten Anteile des Schallsignals aus, während die kleinen Werte von k die niederfrequenten Anteile repräsentieren.
 - Darstellungen von der Form (1) bezeichnet man als **Fourierentwicklung** und sind benannt nach dem französischen Mathematiker und Physiker *Joseph Fourier*, 1768-1830, der solche Darstellungen genauer untersucht hat.
- 

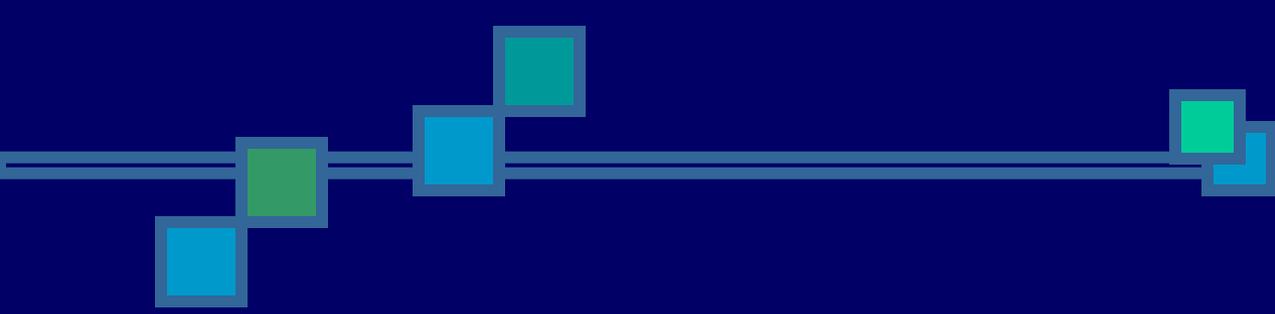


Audio (MP3)

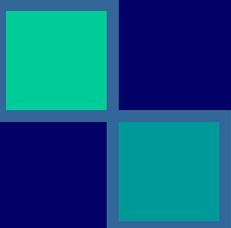
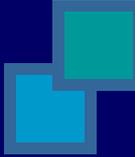


In einem Audiosignal können auch noch höhere Frequenzen als 22 KHz auftreten mit der Konsequenz, dass man diese eigentlich in einer Entwicklung von der Form berücksichtigen müsste. Da solche hohen Frequenzen für das menschliche Gehör nicht feststellbar sind, können diese auch gleich unberücksichtigt bleiben.

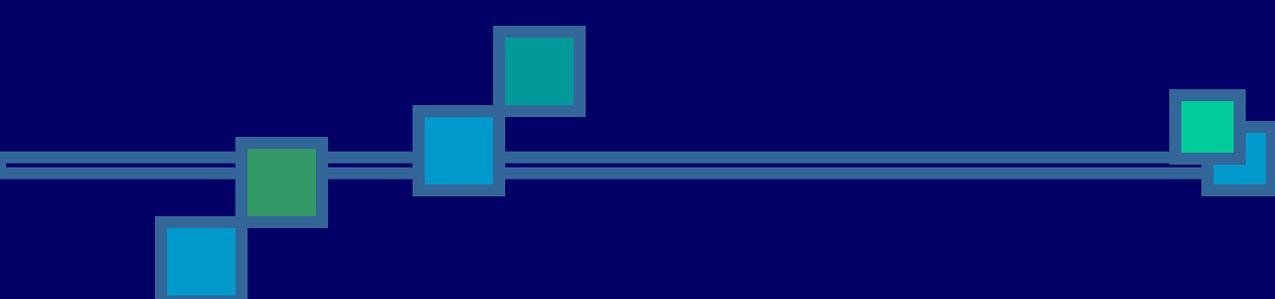




Audio (MP3)

- 
- **Berechnung der Koeffizienten der Sinus- und Cosinusschwingungen**
 1. Gegeben seien die pro Sekunde gewonnenen 44100 Amplitudenwerte
 2. Berechne die 44100 Koeffizienten für die Sinus- und Cosinusschwingung mit Hilfe der schnellen FOURIER-Transformation (Vektor-Matrixmultiplikation)
- 

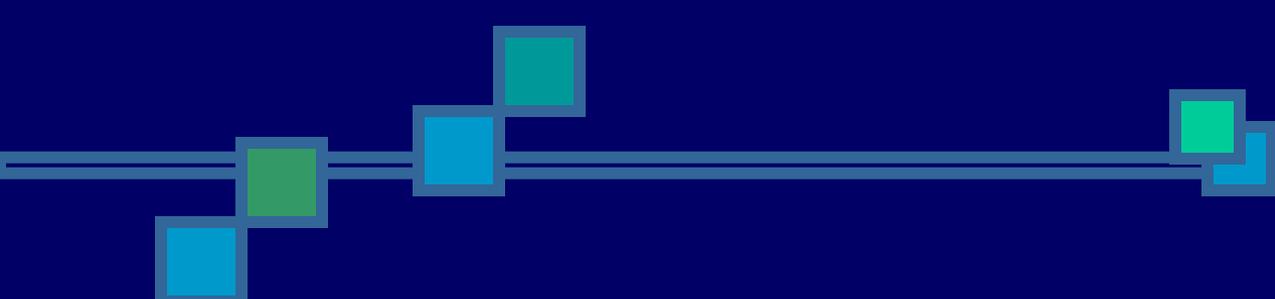
Komprimiere Daten (aber wie?, Lösung folgende Folien)



Audio (Psychoakustik bei MP3)

Filtereinsatz zur Reduktion der Daten

1. Hochfrequente Anteile werden weggelassen, da diese nicht oder nur sehr schwer wahrgenommen werden. In Mp3 werden Frequenzen oberhalb von 16 kHz herausgefiltert.
 2. Tritt bei einer Frequenz ein gegenüber benachbarten Frequenzen sehr lauter Anteil auf, so kann man die benachbarte Frequenz bei der näherungsweisen Rekonstruktion vernachlässigen (MASKIERUNG)
- 



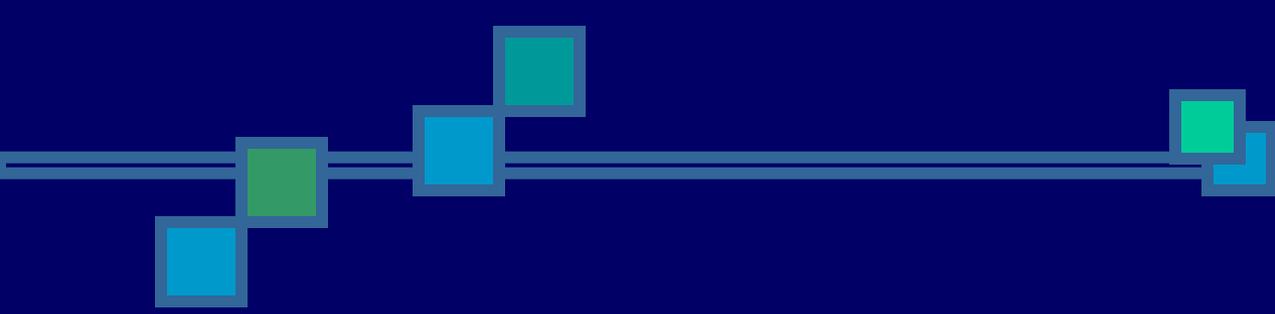
Audio (Psychoakustik bei MP3)

Als Resultat der beschriebenen Vorgehensweise speichert man für jede Sekunde anstelle der 44100 Amplitudenwerte einen gewissen Anteil der auftretenden 44100 Koeffizienten für die reinen Sinus- und Cosinusschwingungen ab. Dafür ist noch eine Quantisierung der Koeffizienten nötig, typischerweise in einer 16Bit-Auflösung.

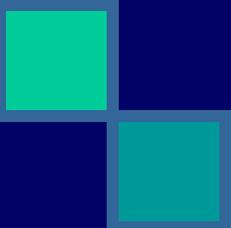
Mit etwa 10 Prozent der gerundeten Koeffizienten, das sind etwa 4400 Amplitudenwerte pro Sekunde, erhält man dabei ein akzeptables Ergebnis. Die zugehörige Bitrate

140 Kilobits/ Sekunde

ist eine typische Größenordnung für das Audiokompressionsformat MP3.



Audio (MP3)



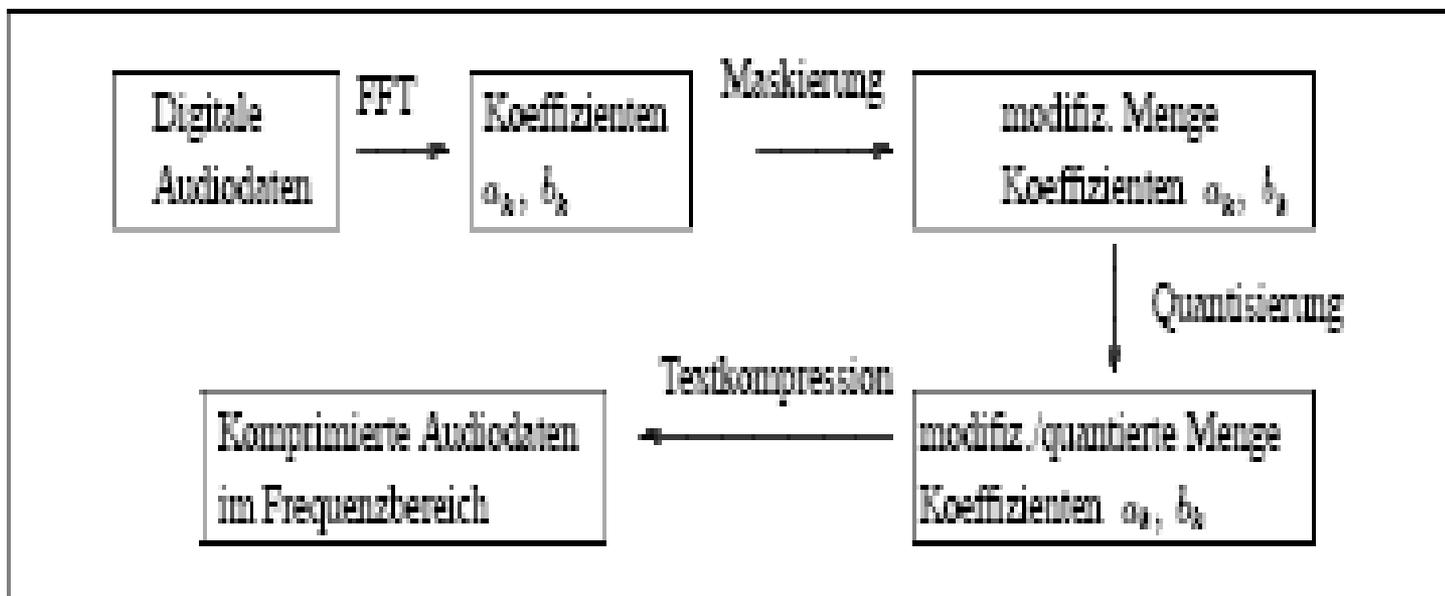
Genauere Beschreibung:

Nico Hartmann: MP3-Grundlagen: Psychoakustik.
(Internet: www.tecChannel.de/multimedia/58/index.html)

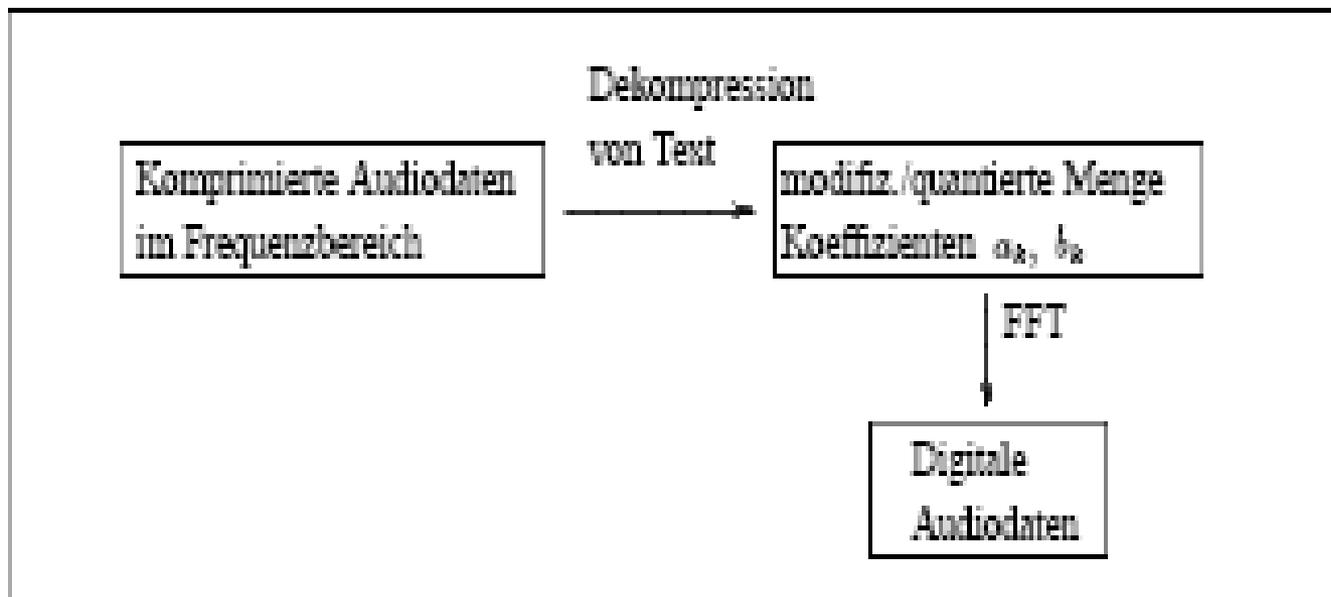


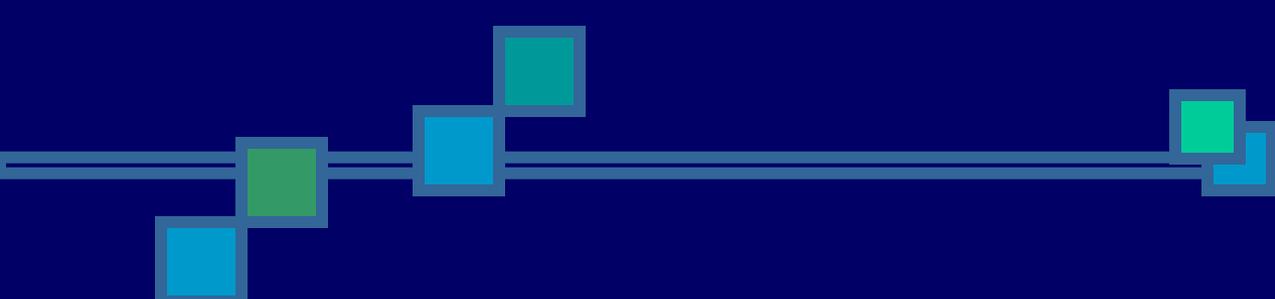
Malte Jeschke: MP3-Grundlagen: Aufbau und Funktion. (Internet: www.tecChannel.de/57/index.html)

Audio (Enkodierung und Kompression bei MP3)

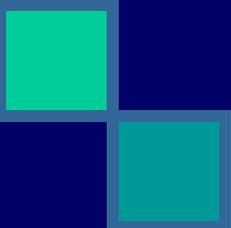


Audio (Dekodierung und Dekompression bei MP3)





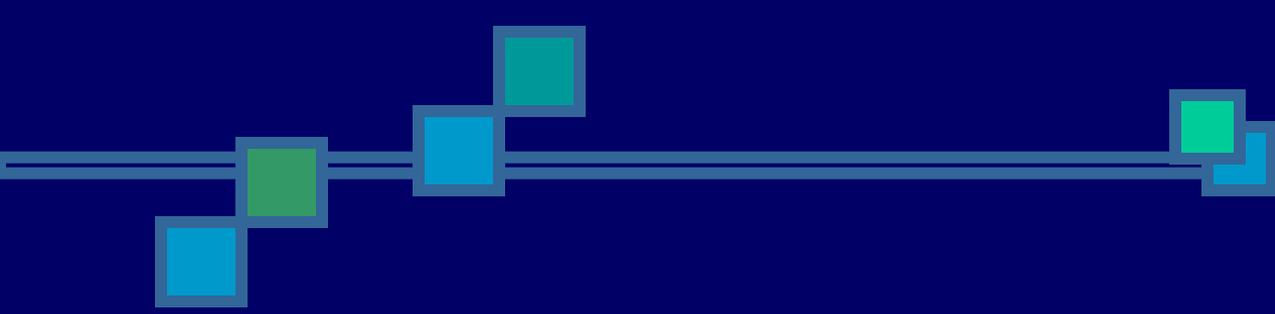
Audio (MP3)

- 
- Test der Software
Mp3 to WAV

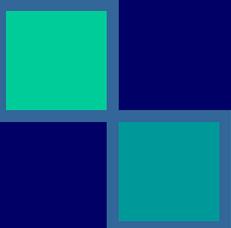
Quelle: www.musica.at

- Test der Software
- 





Audio (MP3)



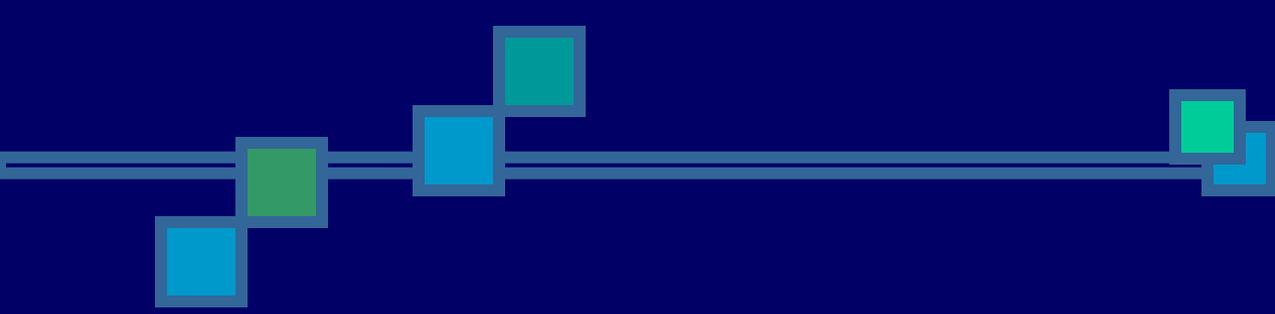
- Weiteres gutes Paket (Umwandlung nach OGG, WMA etc.) unter

www.powerlame.de

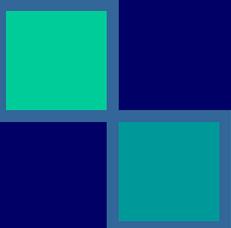
(Freeware)

MP3 Maker von Magix (in der Erprobung durch Studierende)



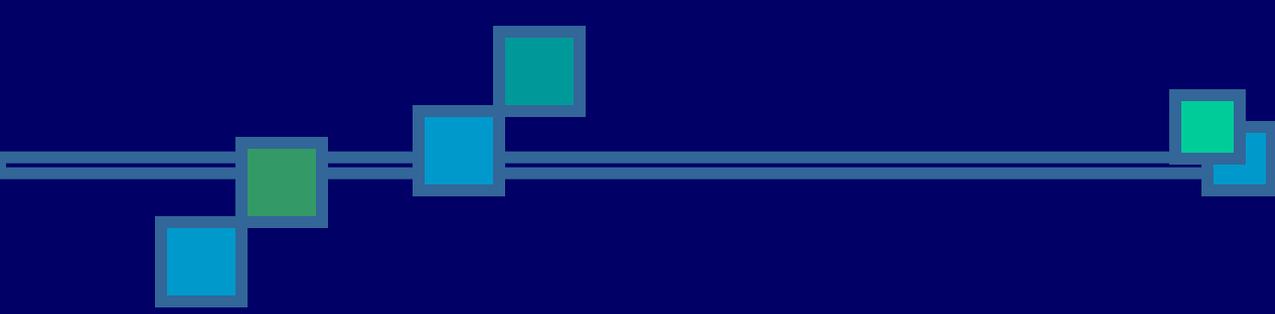


Audio, MP3 und MIDI



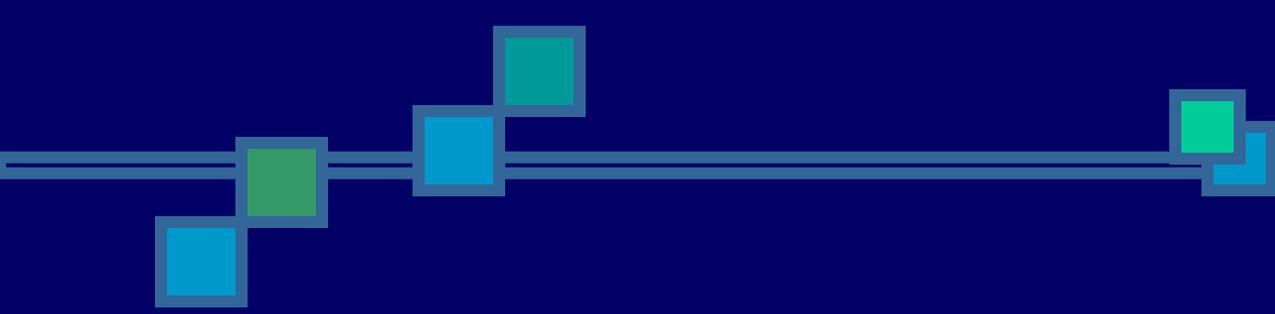
MIDI



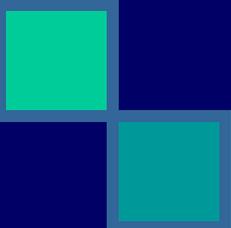
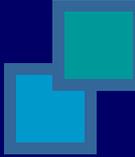


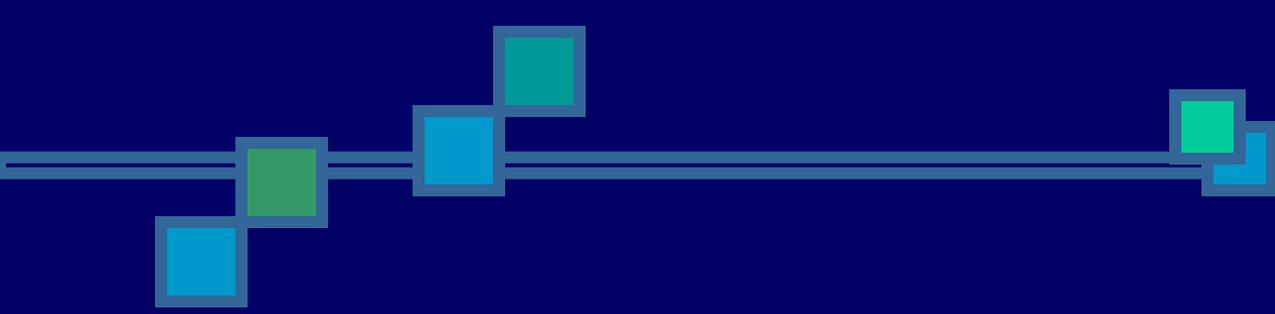
Verwendete MIDI-Literatur

- Huckert, Edgar: Midi-Interpreter Humidi, www.huckert.com
 - Neuschwander, H.W.: MIDI – Musik- und MIDI-Grundlagen. Manuskript FH Kaiserslautern
 - Gorges, P.: Audio, Midi, MP3: Voggenreiter Verlag 2002
 - Schmitz, R.: MIDI. Voggenreiter Verlag 2002
- 

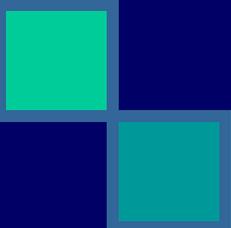


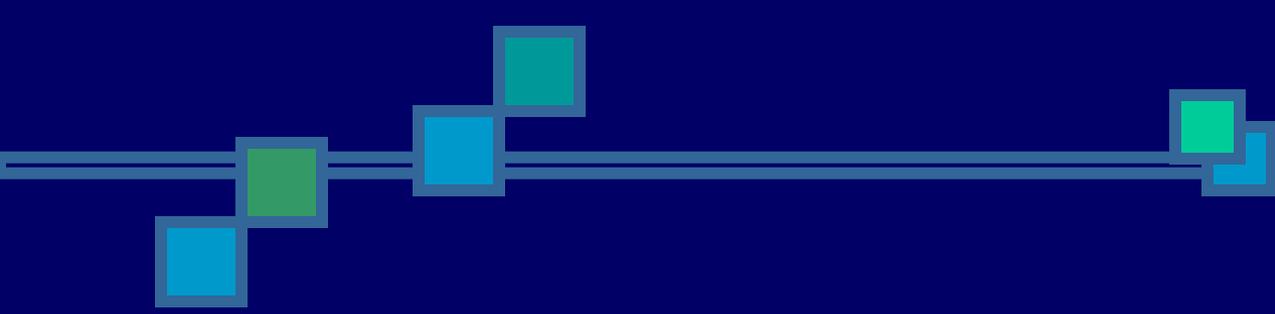
MIDI

- 
- MIDI = Musical Instrument Digital Interface
 - Entwickelt, um mehrere Synthesizer über eine Tastatur spielbar zu machen
 - Heute gültig für Sampler, Drum-Machines, Effektprozessoren, Computer, Audio-Aufnahmegeräte, Geräte zur Lichtsteuerung
 - Gespeichert werden Noten (Zahlen), die Dauer der Note , die Anschlagstärke (extrem kompaktes Datenformat)
- 



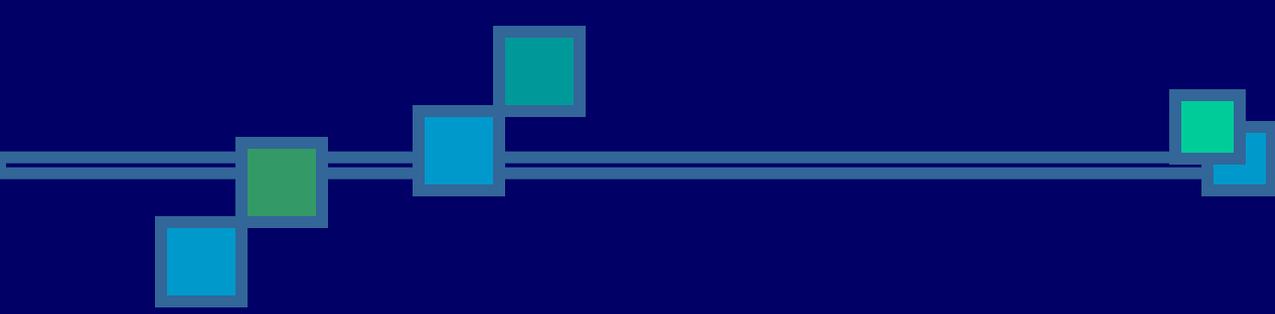
MIDI

- 
- **Beispiel: Bye and Bye (Kenny Burrell, Louis Armstrong)**
 - MIDI-File: 40 KB
 - WAV-File: 29 MB (keine Kompression)
 - WMA-File: 2,7 MB
- 

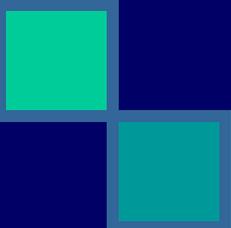


Audio versus Midi

- Audio(WAV, MP3)
 - man nimmt ein konventionell via Mikrophon aufgenommenes Musiksignal und wandelt dies mit einem Sampling genannten Verfahren in eine digitale Wellenform um
 - Midi
 - man greift auf im Computer (oder einem anderen Klangerzeuger) abgelegte, vorgefertigte Klangbausteine zurück, aus denen man mit Hilfe musikalischer Steuerbefehle einen neuen Song im Computer zusammensetzt.
- 



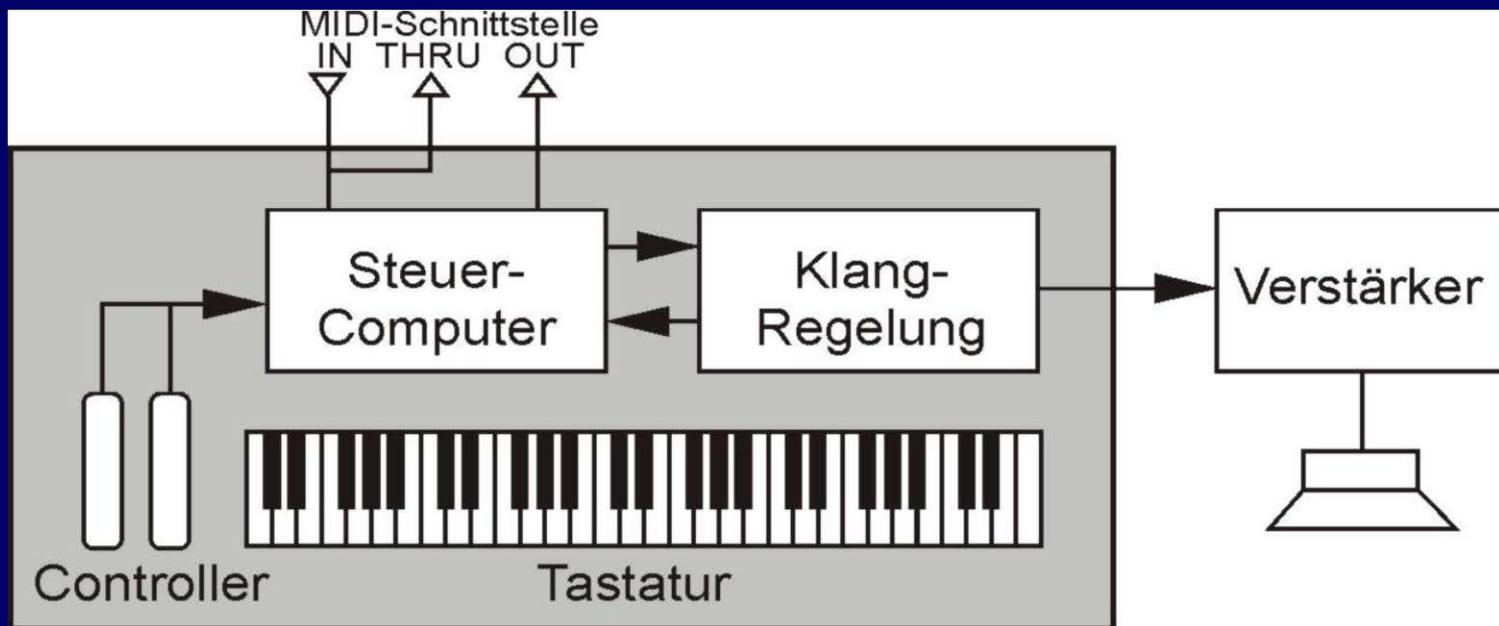
MIDI-Grundlagen

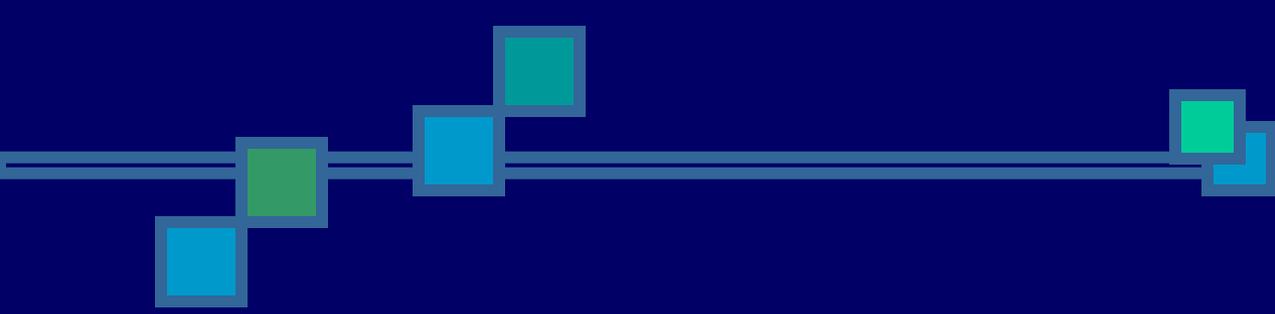


MIDI-Buchsen (IN, OUT, THRU)

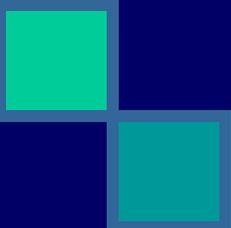
- MIDI-IN dient dem Empfang von MIDI-Daten
 - MIDI-OUT dient dem Versenden von MIDI-Daten, die im Gerät erzeugt werden.
 - MIDI-THRU gibt eine exakte Kopie der Daten aus, die am MIDI-IN des betreffenden Gerätes empfangen wurden.
- 

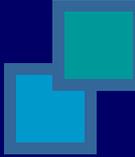
MIDI

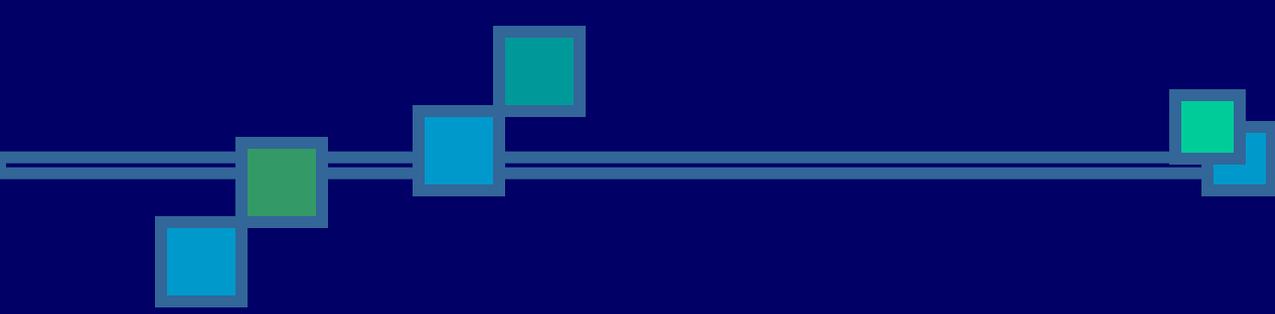




MIDI



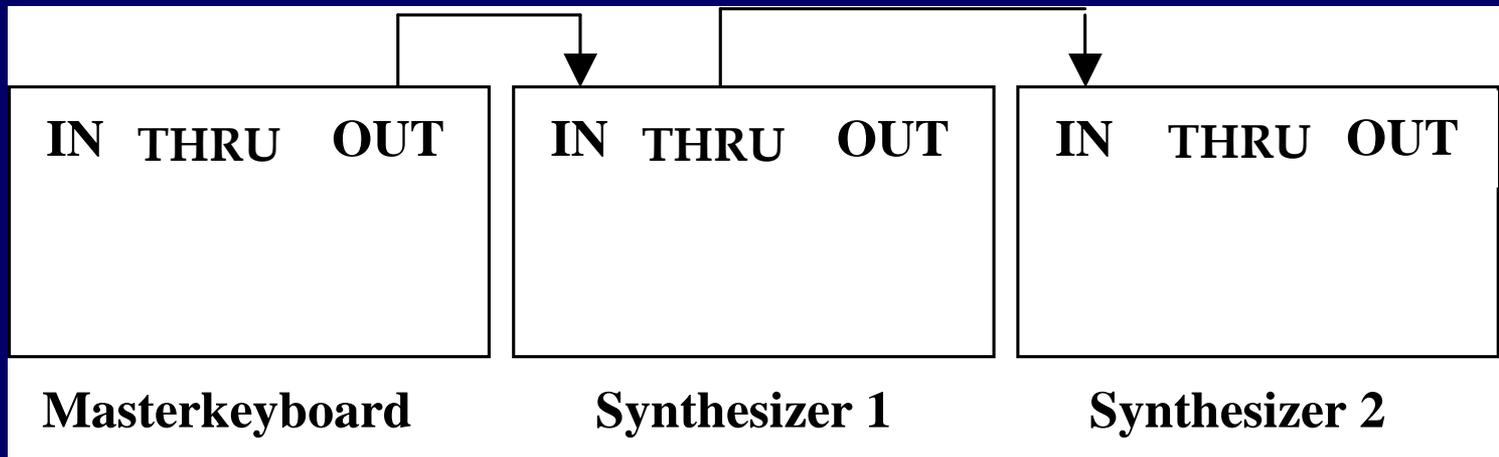
- Die **Tastatur** besteht meist aus vier bis sieben Oktaven. Sensoren unter den einzelnen Tasten leiten Informationen (z.B. die Anschlagstärke) an den Steuercomputer weiter.
 - Der **Controller** kann z.B. ein Modulationsrad, ein Modulationspedal oder ein Sustain-Pedal sein. Controller erzeugen verschiedene Effekte und können auch Klangfarben ändern.
 - Die **Klangerzeugung** kann nach verschiedenen Prinzipien aufgebaut sein, je nach Art des Synthesizers (analoge Klangersynthese, digitale Algorithmus-synthese, Frequenzmodulationssynthese oder Samplingsynthese).
- 

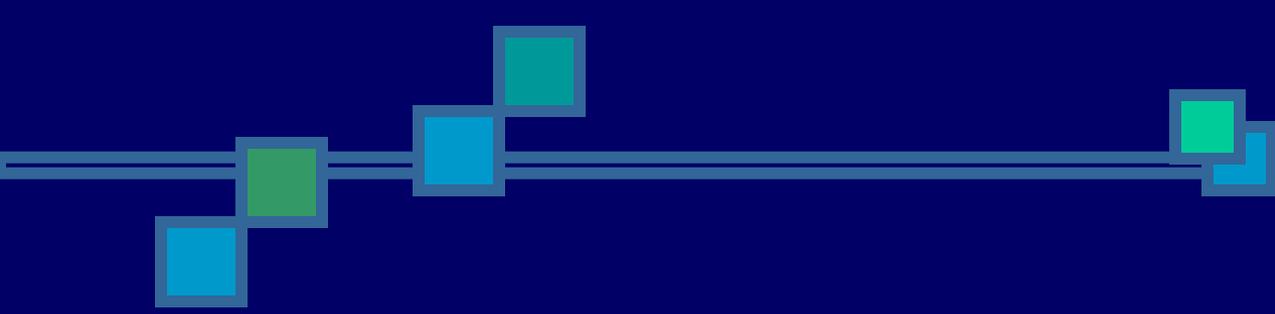


MIDI

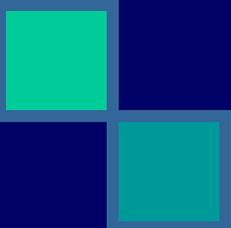
- Der **Klangspeicher** dient zur Speicherung sämtlicher Parameter, die einen Klang beschreiben. Analoge Synthesizer besitzen bis zu 80 Parameter, Synthesizer mit Samplingsynthese besitzen wesentlich mehr (bis zu einige hunderttausend).
 - Der **Steuercomputer** überwacht die Tastatur und die MIDI-Schnittstelle. Ankommende MIDI-Signale werden ausgewertet, durch Tasten, Controller und Schalter erzeugte MIDI-Befehle werden weitergegeben. Er verwaltet den Klangspeicher und stellt die Klangparameter während der Klangbearbeitung ein.
 - Der **Synthesizer** kann über die MIDI-Schnittstelle Informationen mit anderen MIDI-Geräten und mit Computern austauschen.
- 

MIDI-Grundlagen (Verschaltung)



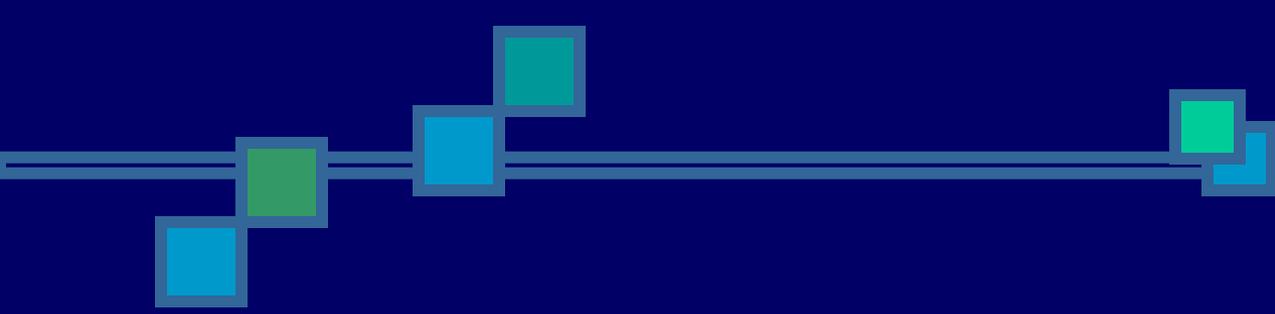


MIDI-Grundlagen



MIDI-Standard unterstützt bis zu 16 Datenkanäle. An jedem dieser Kanäle kann ein Midi-taugliches Gerät angeschlossen werden.





MIDI-Grundlagen



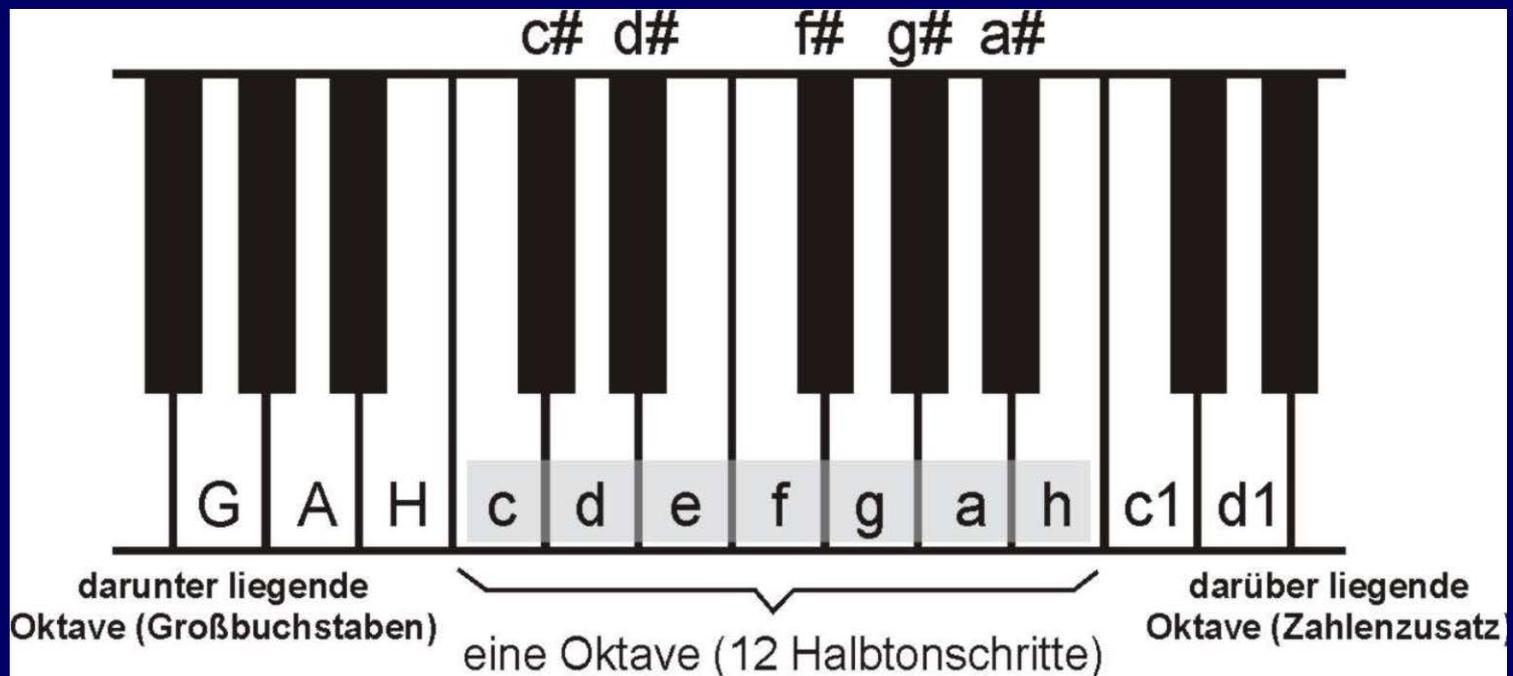
Für das Aufzeichnen von MIDI-Befehlen gibt es ein eigenes Dateiformat, das Standard-MIDI-File (kurz: SMF).

Standard-MIDI-Files (kurz: SMF) gibt es in den Formaten SMF 0, SMF 1 und SMF 2.

Beim Format 0 sind alle MIDI-Kanäle in einer Spur zusammengefasst. Dieses Format wird auch von Klingeltönen für Handys genutzt und kann mit gängigen Sequenzerprogrammen in das Format 1 konvertiert werden. Im Format 1 hat jeder Kanal seine eigene Spur und optional einen eigenen Namen. Verschiedene Stimmen und Instrumente können so besser identifiziert werden. Im Format 2 besteht jede Spur (Track) aus unabhängigen Einheiten. Im Gegensatz zu SMF 1 können also mehrere Spuren dieselbe MIDI-Kanal-Nummer haben.

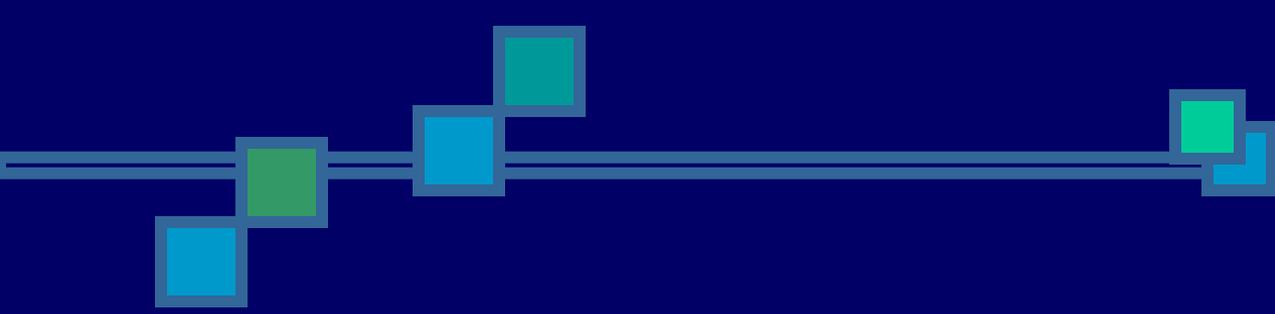


MIDI-Tonnamen und Tonnummern

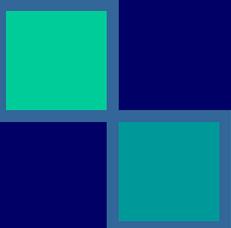


MIDI-Tonnummer	MIDI-Tonname	Musikalischer Notename	Frequenz in Hertz
12	C-2	C2	16,35
13	C#-2	C#2	17,32
24	C-1	C1	32,70
36	C0	C	65,40
48	C1	C	130,81
60	C2	c1	261,62
69	A2	a1 = Kammerton	440
72	C3	c2	523,24
84	C4	c3	1046,50
96	C5	c4	2093
108	C6	c5	4186
120	C7	c6	8372
127	G7	g6	13289

103

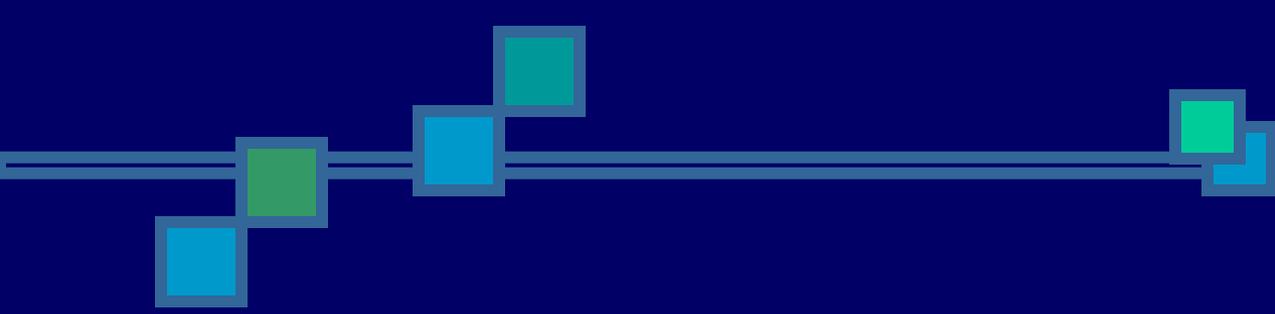


MIDI-Grundlagen (Messages)

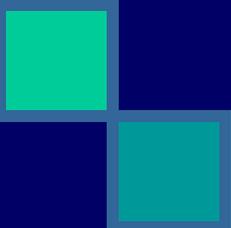


MIDI-Messages sind von den Instrumenten benutzte Kommandos und Daten. Diese beschreiben musikalische Aktionen (=ausführende Aktionen, z.B. Tastenanschlag, Anschlagdynamik, Betätigen von Pedalen, Modulationsräder etc.) und operationale Aktionen (Programmwechsel, Parameterveränderung, Ein- und Ausschalten bestimmter Modes etc.). Eine MIDI-Message ist eine Gruppe von Bytes (8-Bit-Worte), von denen das erste Byte Statusbyte/-wort genannt wird. Datentechnisch sind die 8-Bit-Worte eigentlich 10-Bit-Worte (Start-/Stop-Bit) eingeschlossen. Start-Stop-Bit hat den Wert 0.





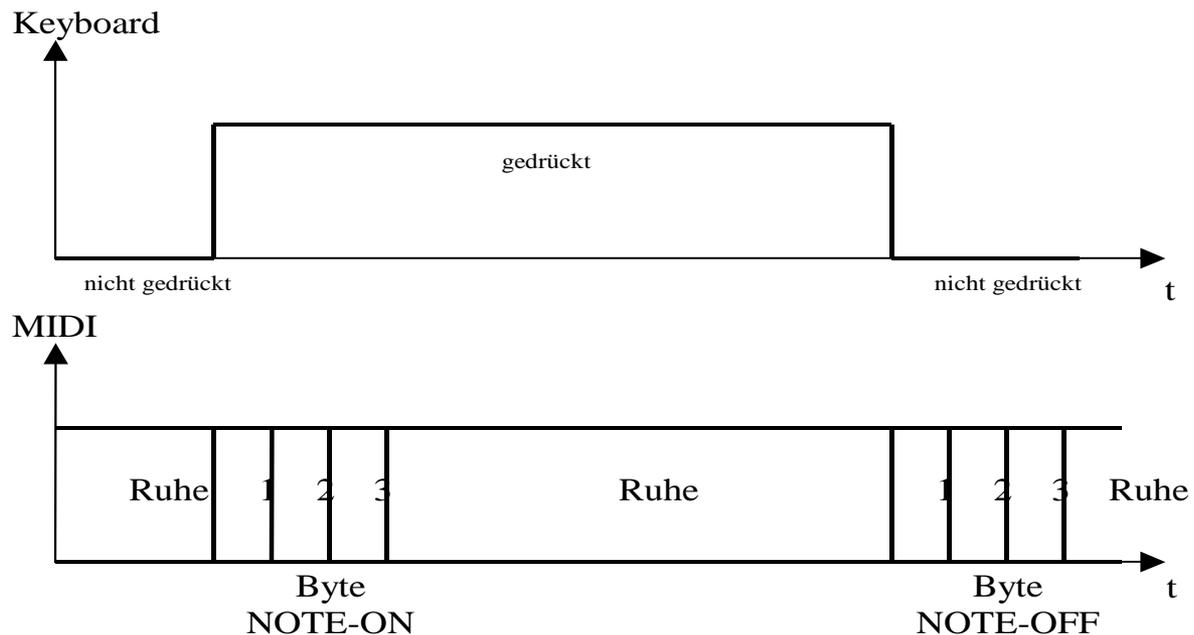
MIDI-Grundlagen (Messages)



MIDI-Messages bestehen aus:

- 1 Statusbyte alleine \Rightarrow Einbyte-Message
 - 1 Statusbyte + 1 Datenbyte \Rightarrow 2-Byte-Message
 - 1 Statusbyte + 2 Datenbyte \Rightarrow 3-Byte-Message
 - 1 Anfangs-Statusbyte + „n“ Datenbyte + 1 End-Statusbyte \Rightarrow n-Byte-Message (z.B. System-Exclusive-Message)
- 

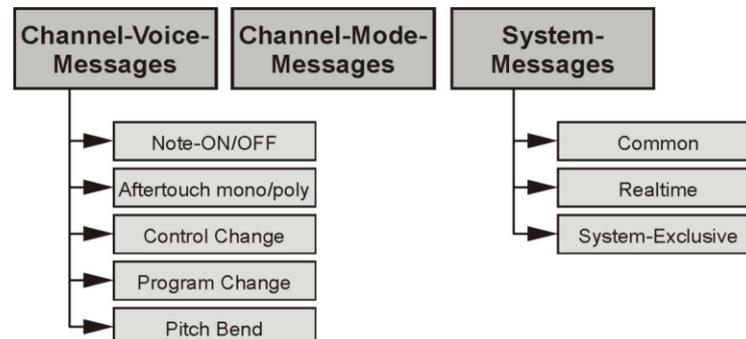
MIDI-Grundlagen

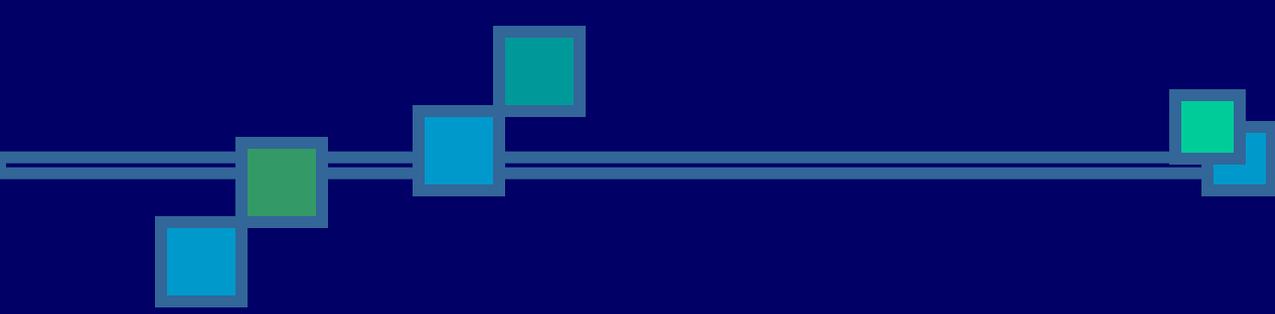


MIDI-Grundlagen

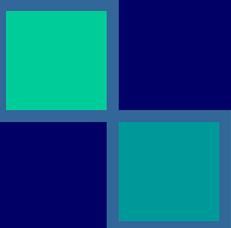
MIDI-Messages werden in folgende drei Gruppen unterteilt:

- *Channel-Voice-Messages*
- *Channel-Mode-Messages*
- *System-Messages*





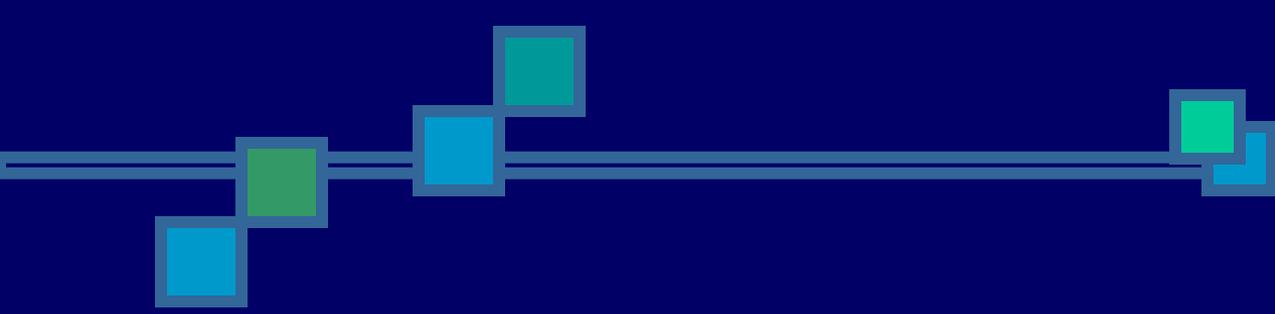
MIDI-Grundlagen



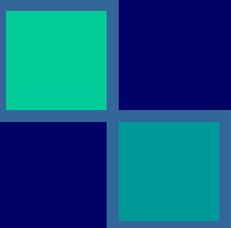
Channel-Voice-Messages:

Channel Voice-Messages bestehen aus einem Statusbyte und einem oder 2 Datenbytes. Sie dienen zum Ein- und Ausschalten von Klängen auf dem Tonerzeuger, zur Anwahl neuer Klänge sowie für Klangeffekte. Sie sind immer auf einen von 16 möglichen MIDI-Kanälen bezogen, die Nummer des Kanals ist im Statusbyte in den letzten 4 Bit enthalten.



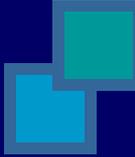


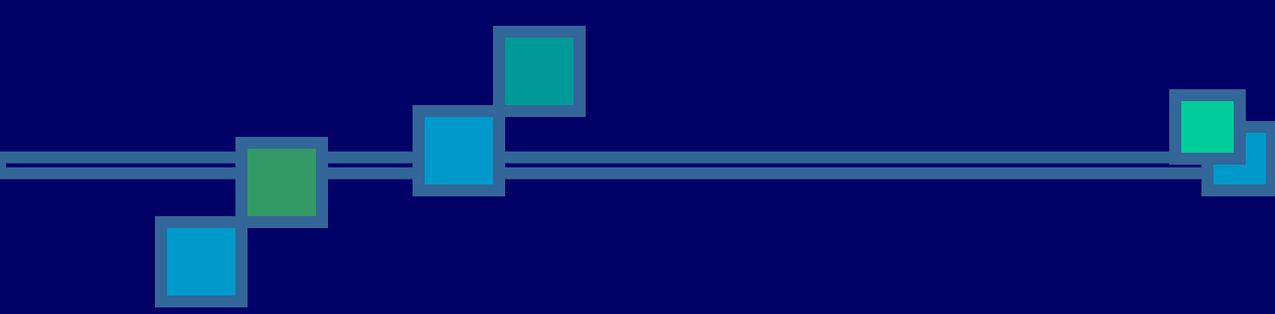
MIDI-Grundlagen



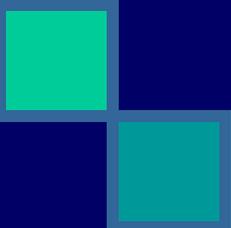
Channel-Mode-Messages:

Channel-Mode-Messages sind eigentlich nur spezielle „Control-Change-Messages“ (3-Byte) und übertragen Betriebsarteneinstellungen, welche im Normalbetrieb selten gesendet werden (z.B. „Local-Off“ => Tastatur an einem Keyboard deaktivieren => nur der Klangerzeuger im Keyboard arbeitet noch).





MIDI-Grundlagen



System-Messages:

System-Messages bestehen aus einem oder „m“ Byte (1 Statusbyte am Anfang + „n“-Datenbyte + 1 Statusbyte am Ende).

- ⑩ *Common-Messages* bestehen aus einem Byte und steuern z.B. den Einsatz eines Sequenzers (Start, Stop usw.). Ein Sequenzer spielt im Prinzip ein Musikstück wie ein Tonband ab mit dem Unterschied, daß dieser nur MIDI-Messages zu einem Klangerzeuger schickt und dieser die Töne erzeugt.
- ⑩ *Realtime-Messages* verschicken Nachrichten, welche möglichst schnell bei einem Empfänger ankommen müssen und dort einen Zeitbezug haben. Ein Beispiel ist der Timing-Clock, welcher zum Synchronisieren mehrerer parallel laufender Sequenzer (oder auch Tonbandgeräte) benutzt wird. Diese Nachricht wird z.B. 24-mal je Note gesendet.
- ⑩ *System-Exclusive-Messages* übertragen im Prinzip komplette Datenblöcke wie z.B. komplette Geräteeinstellungen oder auch Samples.

MIDI-Grundlagen

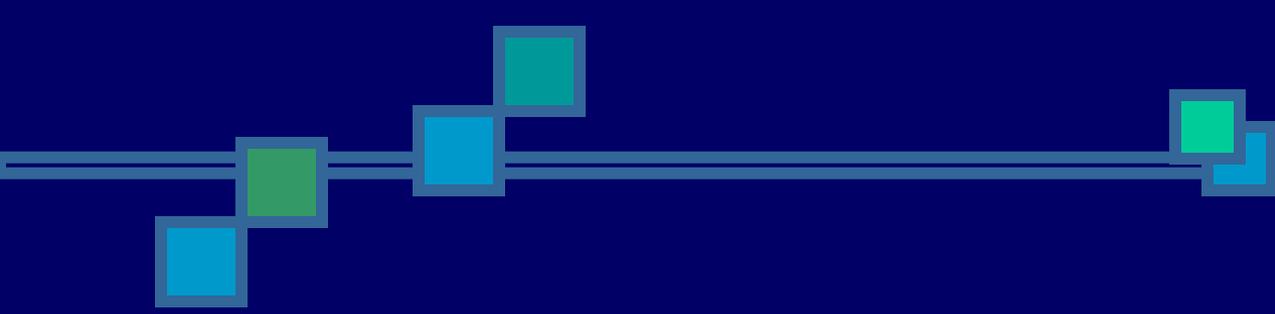
Beispiel NOTE-ON/OFF

Das Ein- oder Ausschalten eines Tones an einem Klangerzeuger erfolgt mit der Message „Note-ON“ sowie „Note-OFF“. Beide sind 3-Byte-Nachrichten.

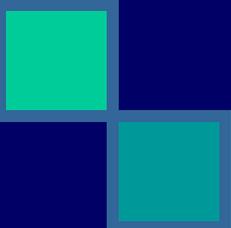
1. Im **ersten Byte (Statusbyte)** ist die Art der Nachricht (Note ON oder Note OFF) sowie die MIDI-Kanalnummer kodiert.

Beispiel:	1001	0101
Bedeutung:	NOTE-ON	Kanal 5

111



MIDI-Grundlagen

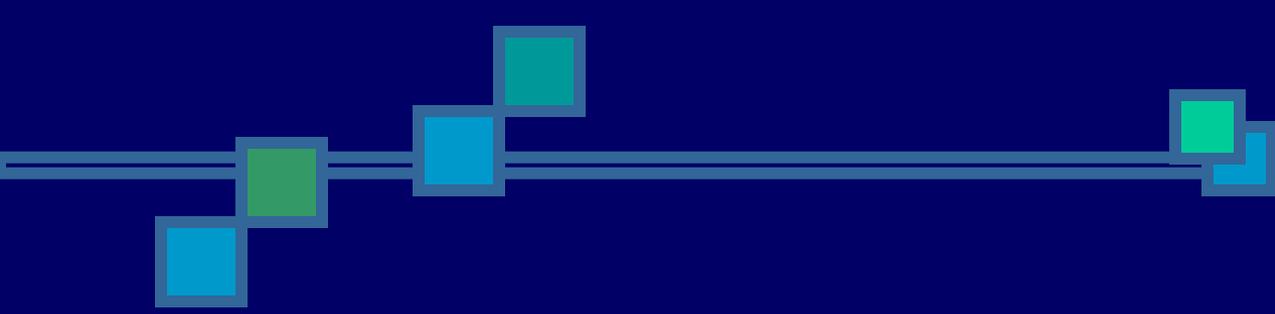


Im **zweiten Byte (erstes Datenbyte)** wird die MIDI-Tonnummer angegeben. Damit wird gleichzeitig indirekt die Frequenz des ein- oder auszuschaltenden Tones festgelegt. Die Frequenz bezieht sich auf die Tonnamen (musikalische Notendarstellung) einer Klaviertastatur. (vgl. Folie 103)

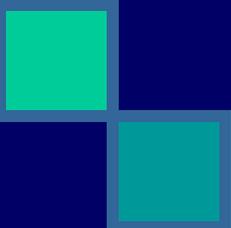


Beispiel: 0 0111100

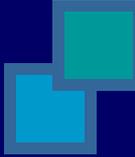
Bedeutung: Im Datenbyte steht an der ersten Stelle der Wert 0, die restlichen 7 Bits dienen der Codierung der Notenummer (hier: 60)



MIDI-Grundlagen



Im **dritten Byte (zweites Datenbyte)** ist die Lautstärke (Velocity) des ein- oder auszuschaltenden Tones kodiert. Es sind Werte von 0 (Ton aus) bis 127_{10} bzw. $7F_H$ (Ton maximal laut) möglich.



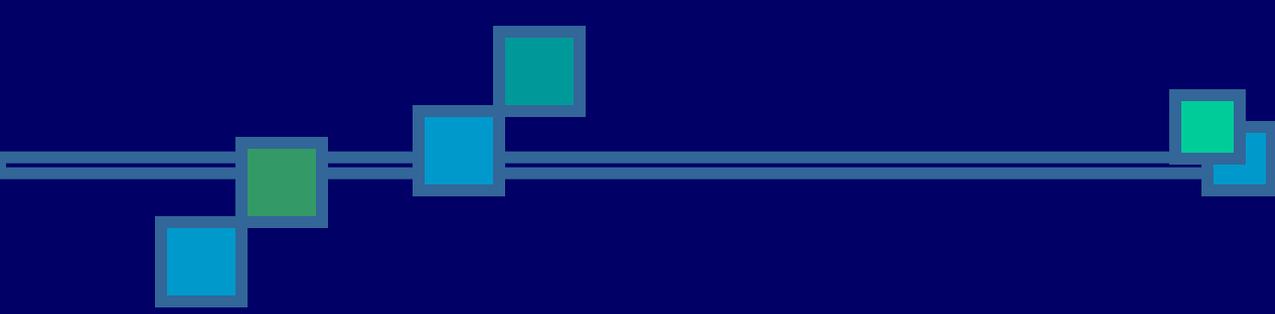
Beispiel: 0 1111001

Bedeutung: Datenbit 1 ist 0, Velocity ist hier 121_{10} oder 79_H

Dies ist sehr laut.

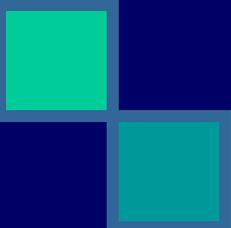
MIDI-Grundlagen

Drittes MIDI-Byte bei der Message "NOTE ON" oder "NOTE OFF"									
0	1	16	32	48	64	80	96	112	127
OFF	pianissimo possibile	pianissimo	piano	mezzo-piano	normal	mezzo-forte		fortissimo	fortissimo possibile
Ton Aus	extrem leise	sehr leise	leise	etwas leise	normal	etwas laut		sehr laut	extrem laut
	<i>ppp</i>	<i>pp</i>	<i>p</i>	<i>mp</i>		<i>mf</i>	<i>f</i>	<i>ff</i>	<i>fff</i>



MIDI-Grundlagen

NOTE-OFF-Kommando



1000 Kanal 1. Datenbyte 2. Datenbyte

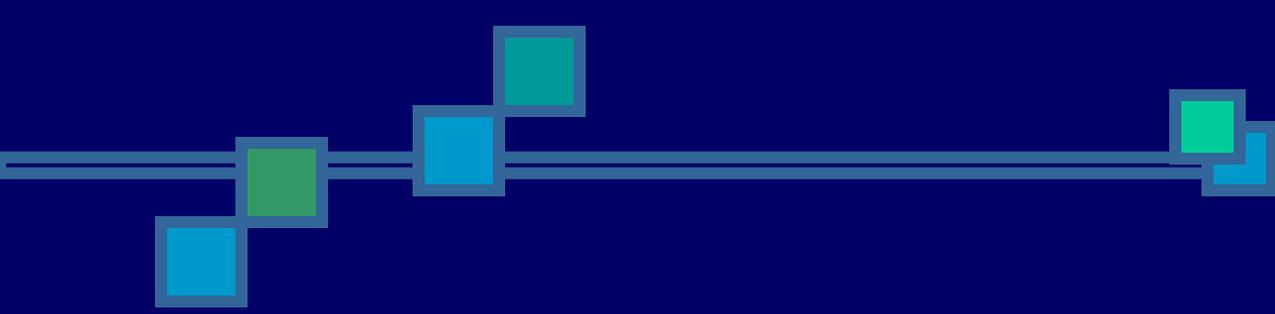
Program Change-Kommando

1100 Kanal 1. Datenbyte 2. Datenbyte

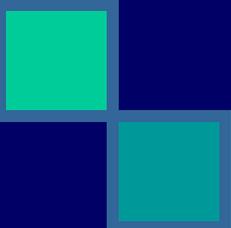
PITCH BEND-Kommando

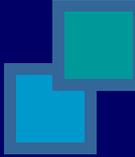
1110 Kanal 1. Datenbyte 2. Datenbyte

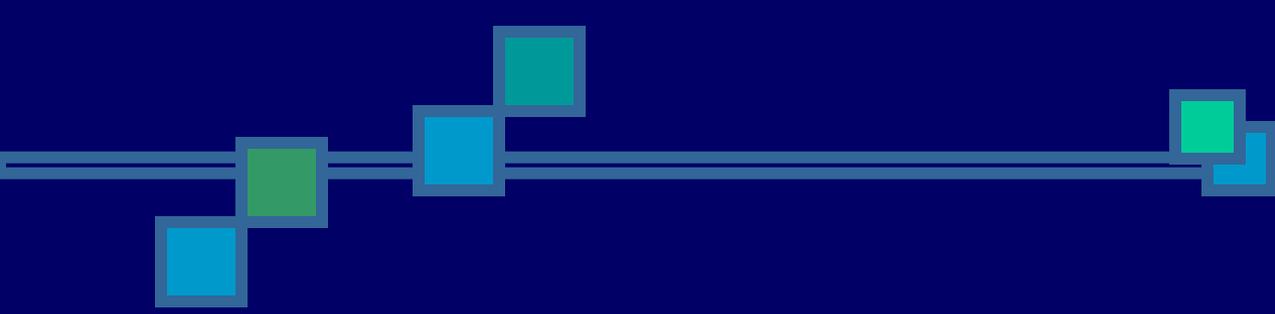




Übungen

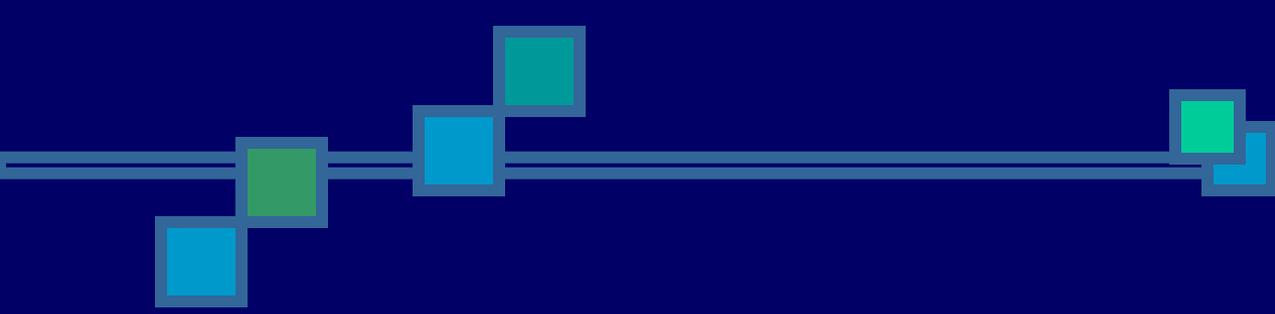


1. Untersuchen Sie den MIDI-File „Blue Bossa“ mit einem Hex-Editor.
 2. Untersuchen Sie den Midi-File „On the sunny side of the street“ mit einem Hex-Editor.
 3. Installieren Sie den MIDI-Interpreter Humidi. Studieren Sie die Hilfe. Untersuchen Sie die Datei „Blue Bossa“ und „on the sunny side of the street“.
- 

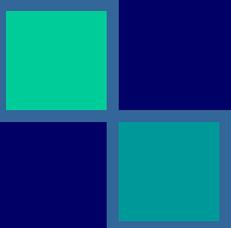
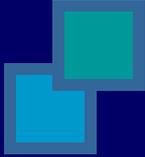


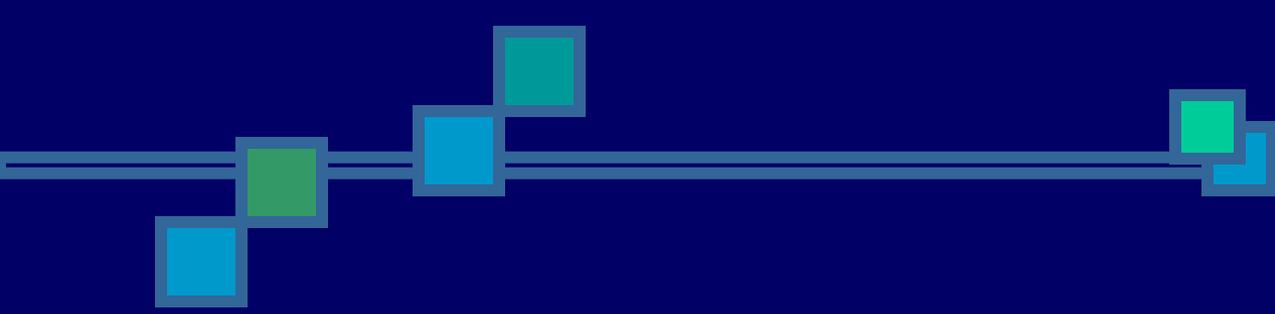
Digitalformate

- Übertragungsformate (Standards zur Echtzeitübertragung von Daten zu digitalen Audiosystemen)
 - AES/EBU
 - S/D-DIF
 - ADAT Lightpipe (ADI)
 - Tascam TDIF
 - Firewire (IEEE 1394)
- 



Software für Musik

- Notation und Arrangement
 - Lernsoftware (Klavier, Gitarre, Rhythmus, Gehörbildung)
 - Audio-Bearbeitung und MP3
 - Sequenzer
 - MIDI
 - Datenbank Anwendungen
- 
- 

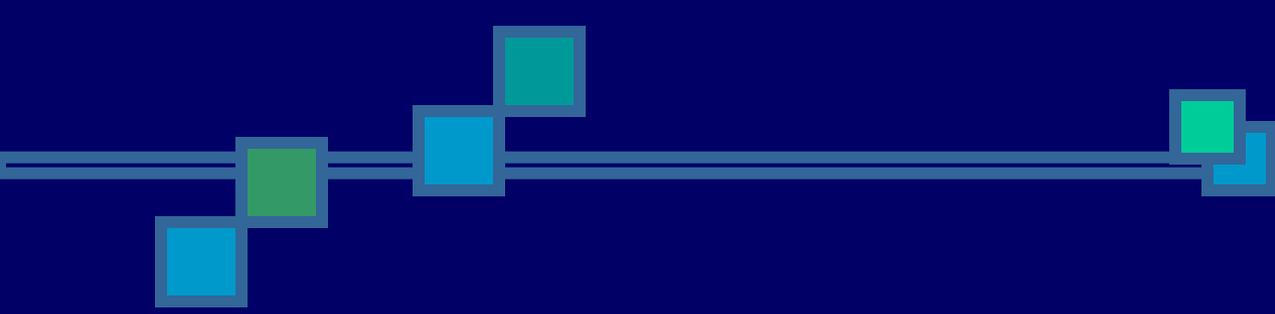


Sequenzler

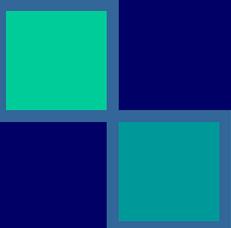
- Definition:

Ein Programm oder eine Hardware, welche in bestimmten festgelegten Zeitabständen Befehle (z.B. Informationen über zu spielende Töne und Instrumentenstimmen) an elektronische Musikinstrumente abgibt und somit über der Zeit ein Musikstück abspielen kann. Beispiele einfacher Sequenzer sind z.B. Leierkästen oder mechanische Klaviere mit Abspielvorrichtung, bei welchen die Musik in Lochform auf Papierstreifen aufgebracht ist.



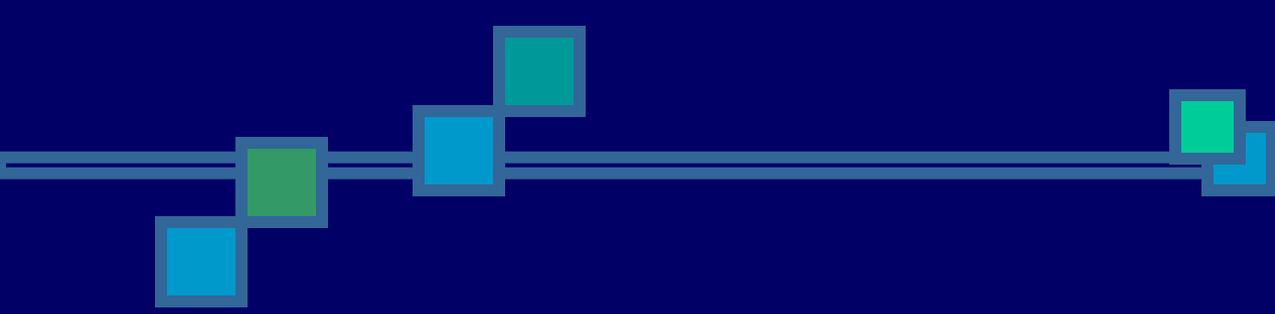


Sequencer



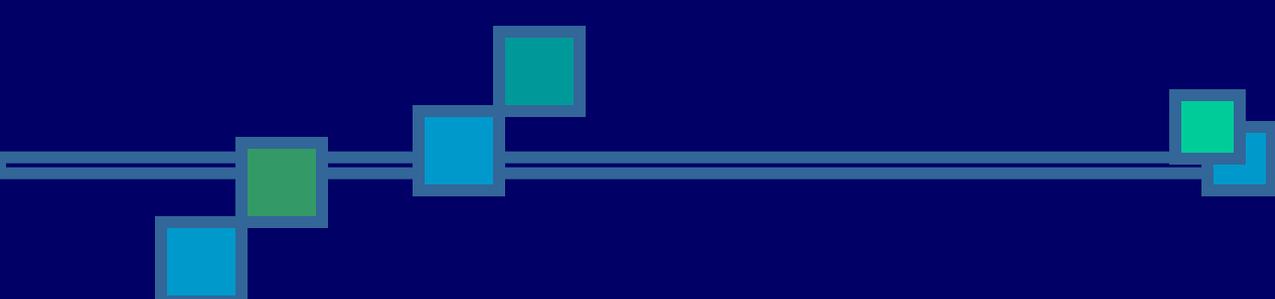
- Einsatzgebiete

- Mischen von MIDI- und Audio-Spuren zu einem neuen Song
 - Komponieren von neuen Songs
 - Ausdruck von Noten
- 



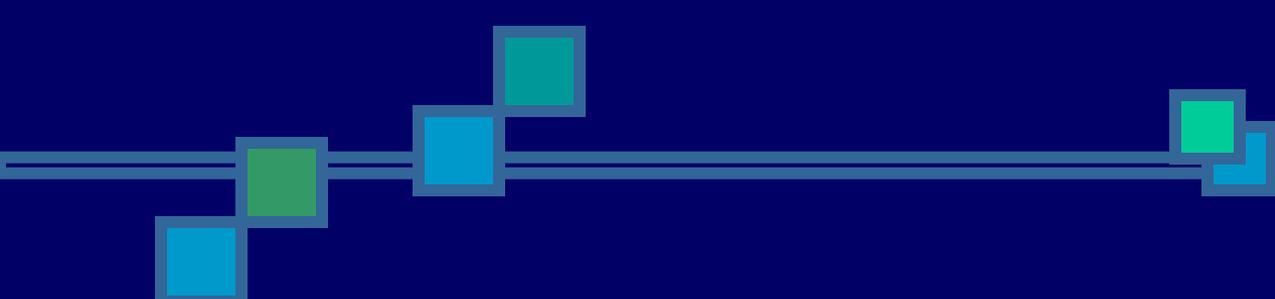
Software Sequenzer

- Cubasis VST (Pinnacle)
- CUBASE (Steinberg = Yamaha)
- Logic (Emagic)
- Cakewalk (Cakewalk)
- Digital Performer (MOTU)
- Vision (Opcode)
- Pro Audio (PowerTracks)



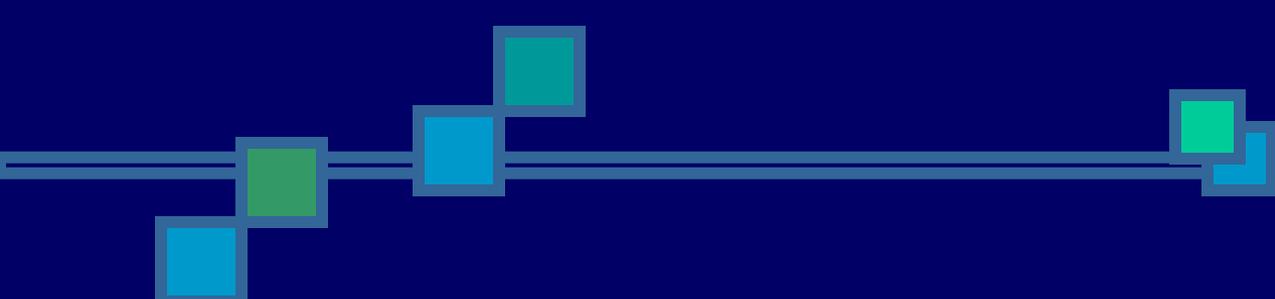
Grundfunktionalität Sequenzersoftware

- eine zentrale Bedieneroberfläche (*Arrangierfenster*), auf der die eingespielten Tonfolgen bzw. Sequenzen als waagerechte Balken erscheinen und mit der Maus beliebig hin und her verschoben werden können
 - ein Feld zur Steuerung des Sequenzers (*Transportfeld*), das mit Icons versehen ist, die den Laufwerkstasten eines CD-Players ähneln
- 



Grundfunktionalität Sequenzersoftware

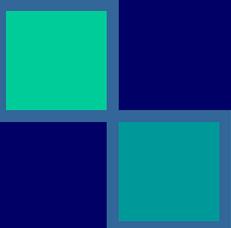
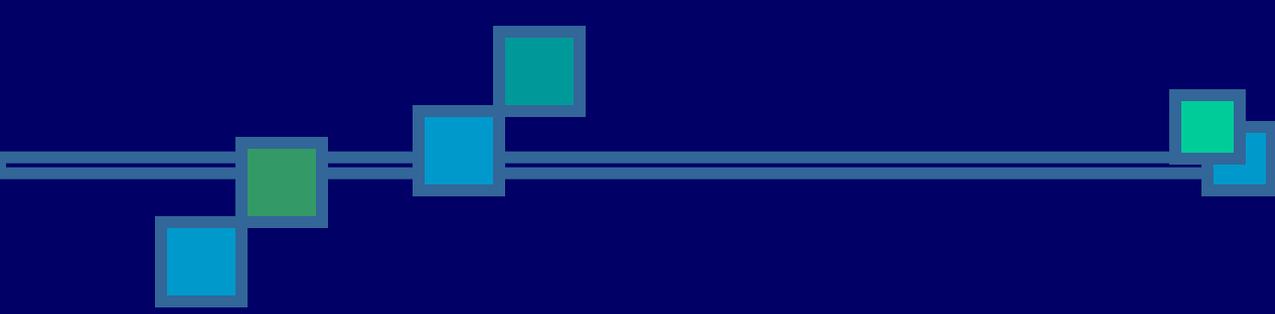
- ein virtuelles Mischpult (*MIDI-Mischer*), das, wie sein realer Kollege, Drehknöpfe und Fader besitzt, und mit denen man den Zusammenklang mehrerer Instrumente beeinflussen kann
 - weitere Unterprogramme, die so genannten Editoren (*Noten-Editor, Matrix-Editor* usw.), die z.B. die Noteneingabe erleichtern oder für Ausbesserungsarbeiten verwendet werden
- 



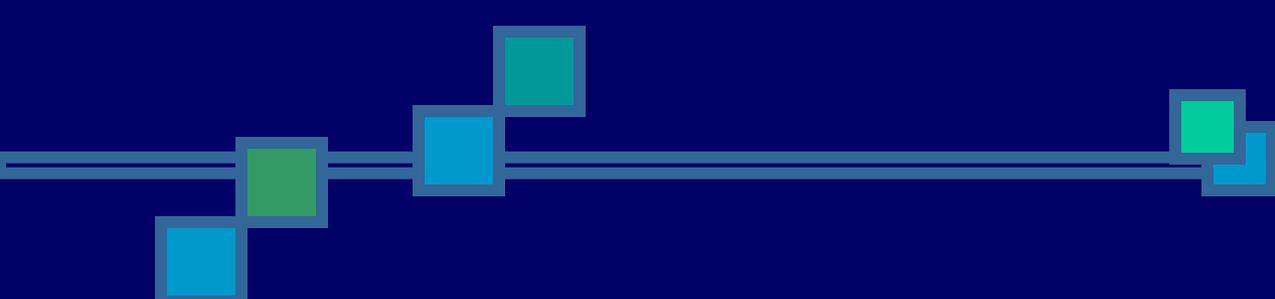
Grundfunktionalität Sequenzersoftware

Demonstration von Sequenzersoftware und
Mischsoftware (LOGIC) im Tonstudio
Blaufabrik Hurch B. Wittmann





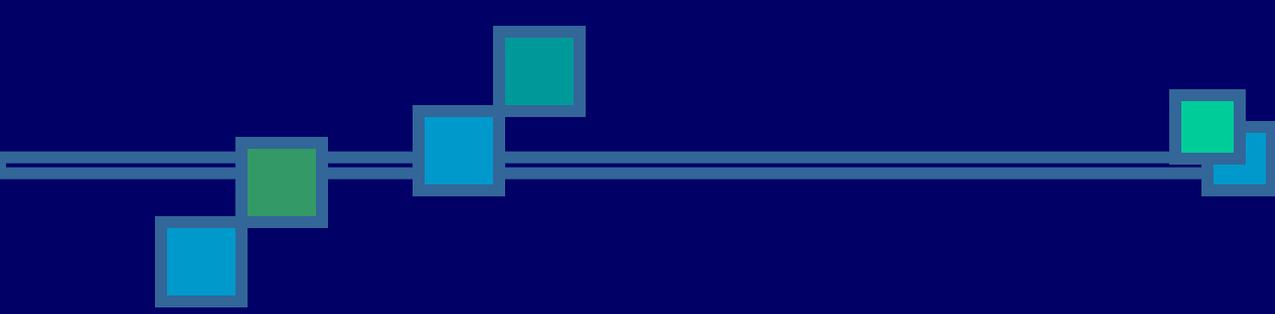
Üben, Arrangieren, Komponieren, Mischen und Mastern



Üben, Arrangieren, Komponieren, Mischen, Mastern

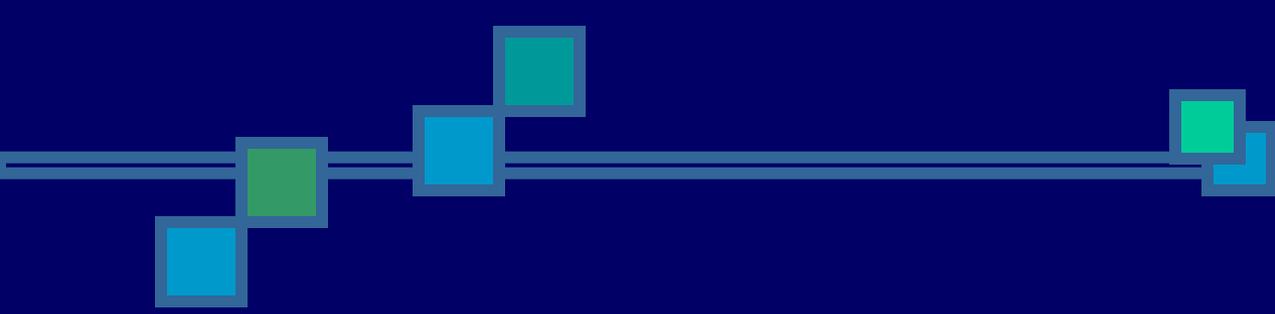
■ Üben

- Lehrhefte
 - Play Along z.B. Aebersold im Jazzbereich
 - Lehrvideos
 - DVD
 - Software (z.B. Band in a box)
- 

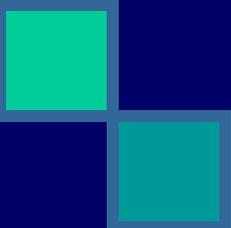


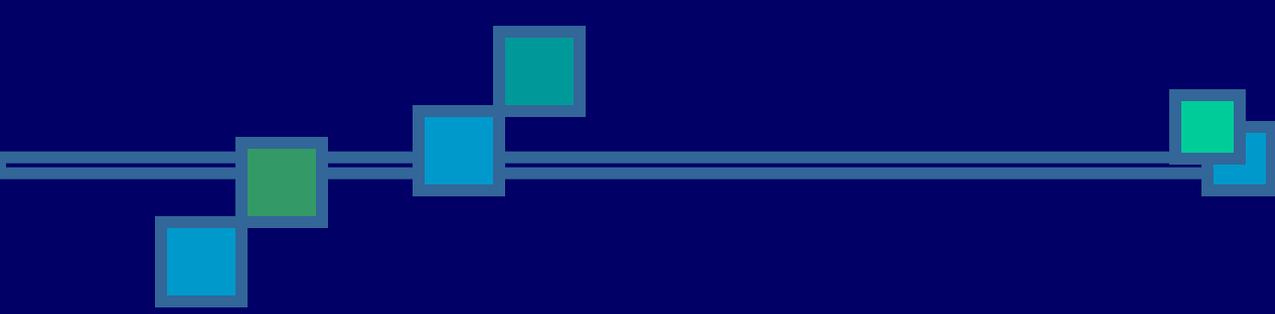
Überblick Band In A Box (1)

- Play Along Funktionalität durch Eingabe von Akkorden und Auswahl des Styles
 - Import von Midi Files und Neuarrangement
 - Unterstützung von MIDI-Instrumenten
 - Generierung bzw. Aufsatz auf Akkordschema von Soli in bestimmten Stilrichtungen/Interpreten
 - Improvisationsschemata
- 



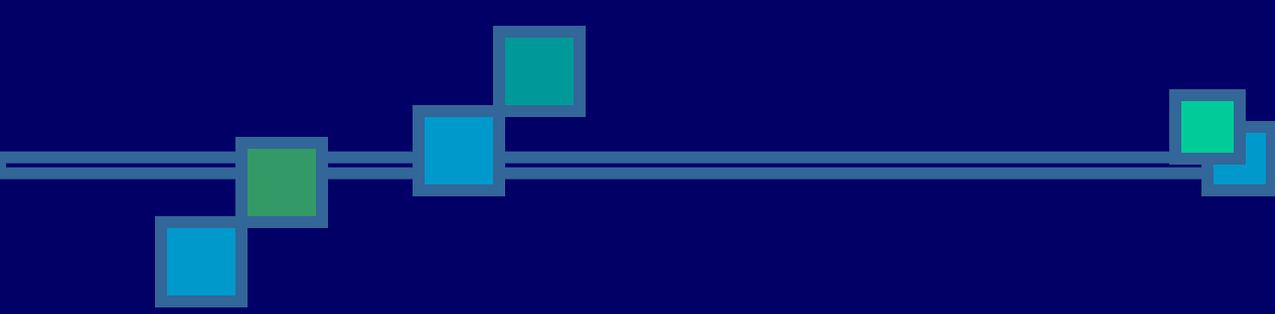
Überblick Band In A Box (2)

- 
- Herstellung von Midi-/Audio-Files für Play-Along oder Live-Darbietungen
 - Lernsystem für diverse Instrumente
 - Notationen für unterschiedliche Instrumente
 - Stimmfunktionen
- 

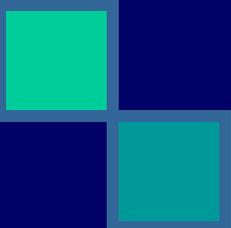


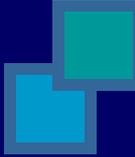
Praktikum 3
BAND IN A BOX

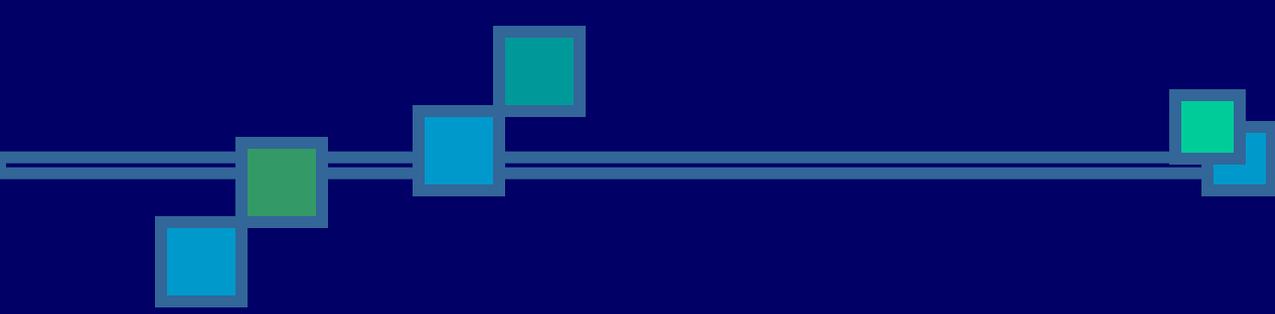




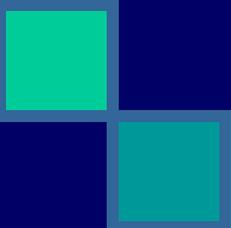
Praktikum

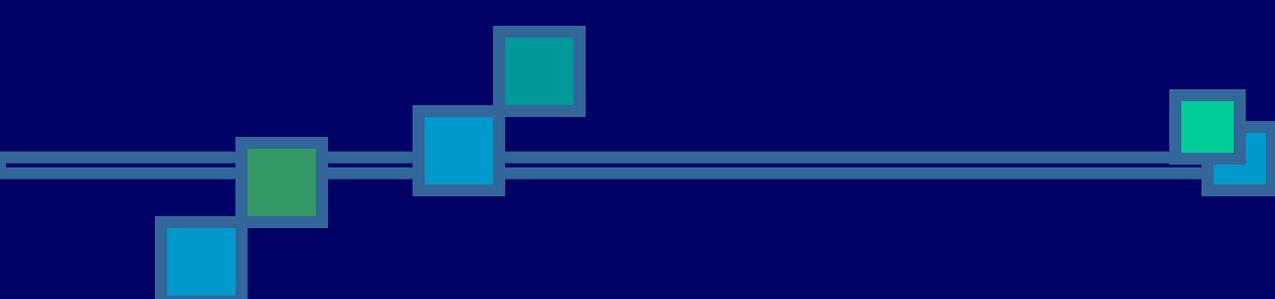


1. Erfassen Sie die Akkorde mittels BB von „On the sunny side of the Street“ und speichern Sie diese.
 2. Öffnen Sie den mit Print MUSIC! erzeugten Midi-File „On the sunny of the street“
 3. Interpretieren Sie Unterschiede
 4. Erstellen Sie eine Audio-Datei mit der unter 1. erzeugten Version.
 5. Erzeugen Sie mit der unter 1. dargestellten Version eine Soloisten-Datei.
- 



Praktikum

- 
6. Erzeugen Sie für „On the sunny side of the street“ eine Melodie-Datei im Stil „Swing Vibraphon“
 7. Erzeugen Sie für die unter 6. generierte Datei einen Audio-File.
 8. Drucken Sie die Melodiestimme für die unter 6. erzeugte Datei
 9. Bearbeiten Sie die unter 6. generierte Melodie
- 

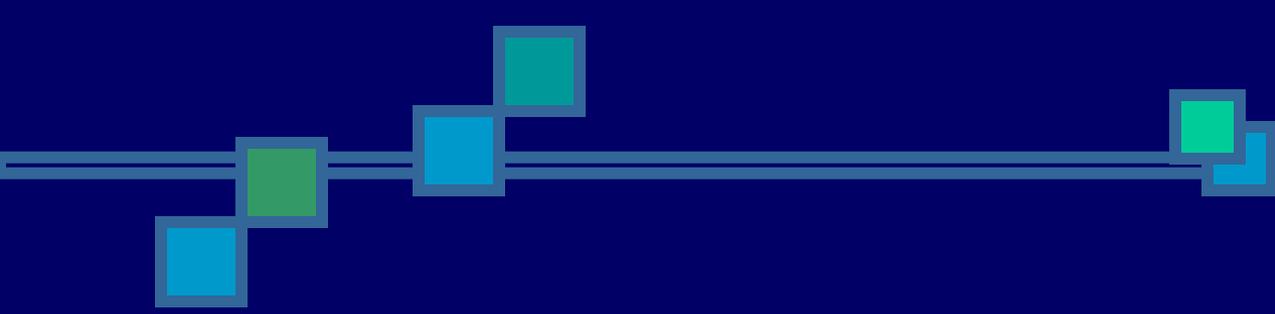


Abschlusspraktikum Band in a box

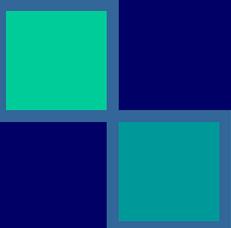
- Vorbemerkung:
 - Auf Laufwerk Q liegen im Verzeichnis „Schreibrecht für alle“ die BB-Dateien „Blue Skies“ und „Route 66“.
- 

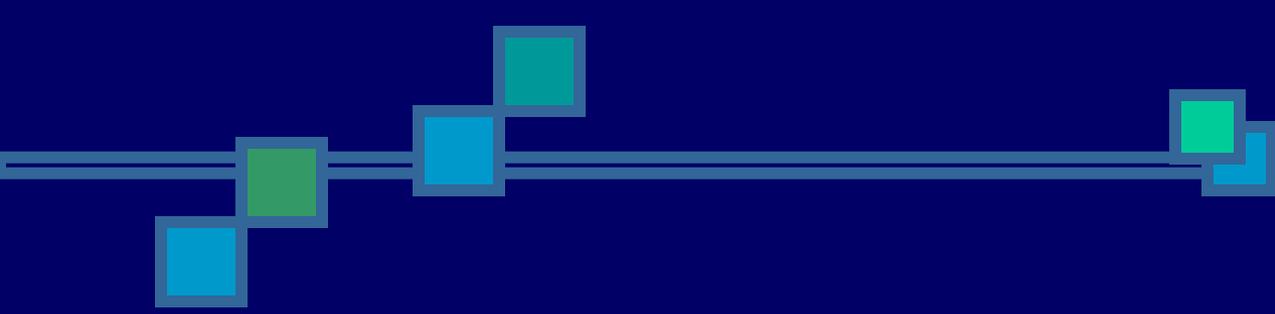
Abschlusspraktikum Band in a box

- Laden Sie die genannten Dateien und speichern Sie die Dateien mit Ihrem angehängten Familiennamen ab. Arbeiten Sie nur mit diesen Dateien.
- Drucken Sie sich die Melodiestimmen mit Akkorden aus
- Transponieren Sie die Stimme für Bb-Instrumente.
- Legen Sie einen anderen Stil (passend) auf die Melodiespur.
- Speichern Sie die Datei im Audioformat WAV und testen Sie das Ergebnis mit einem Audioplayer.
- Erzeugen Sie MIDI-Dateien für beide Stücke.
- Erzeugen Sie aus der MIDI-Datei einen WAV-File. Vergleichen Sie die Ergebnisse mit dem unter 5. erzielten Ergebnis.
- Erzeugen Sie mit Hilfe des Soloisten eine neue Melodie für beide Dateien
- Legen Sie auf die WAV-Dateien passend Chorus, Delay und Flanger.

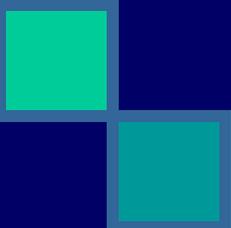


Zusatzaufgabe: Techno-Styleset Nr. 69
im Verzeichnis für Band in a box. Analy-
sieren sie mindestens zwei Melodien.





Gastvortrag



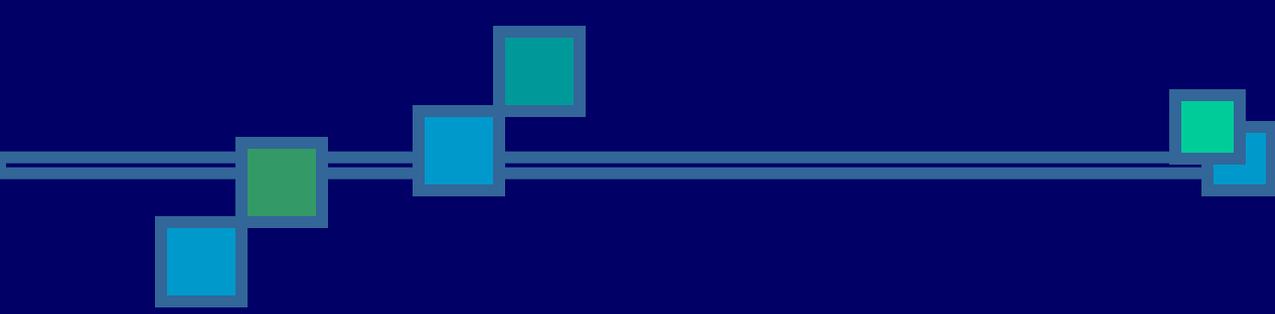
**Improvisation in Blues und Jazz
mit Computerunterstützung**

durch

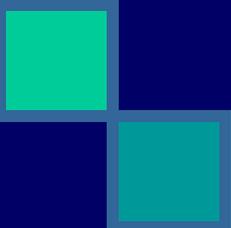


Martin Schiel

21.6.2010

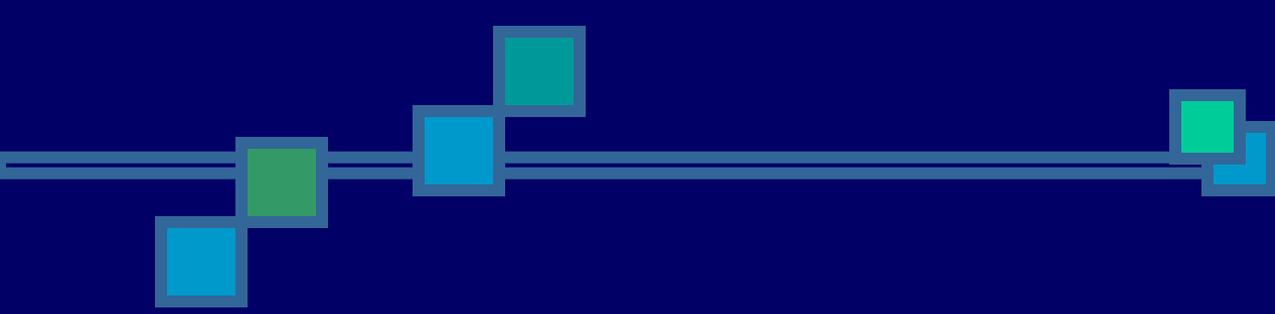


Gastvortrag

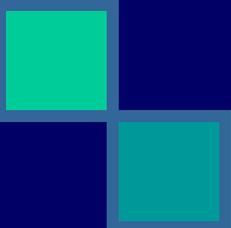


Themen:

- Melodiespiel
 - Grundlagen Improvisation
 - Einfache Improvisation unter Verwendung von Band in a box (BB) (Moll-Blues, Dur-Blues)
 - Improvisation über „Softly“ unter Verwendung BB
 - Fortgeschrittene Improvisation in Swing und Latin (On the sunny side of the street/All the things you are/Blue Bossa) mittels BB
 - Diskussion
- 

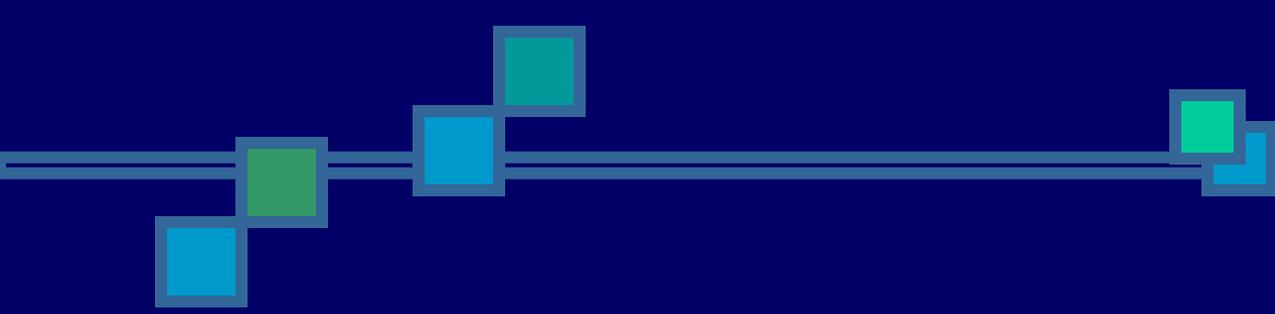


Virtuelle Studio Technologie

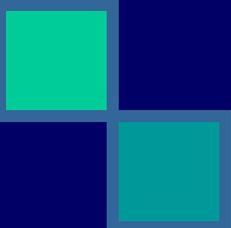


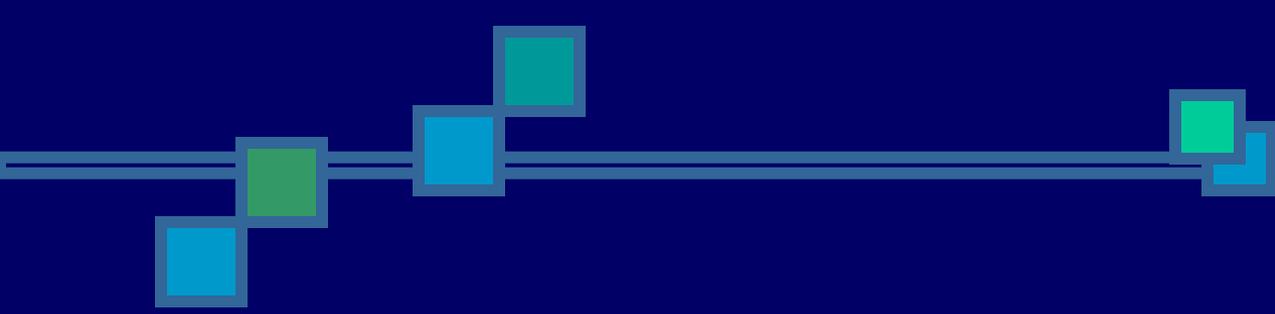
■ **Historie**

- Experimente durch Tangerine Dream, Kraftwerk, La Düsseldorf, Neu, Jean Michel Jarre
 - 1973 kommerzieller Einsatz von Synthesizern in der Produktion von Hot Butter bei "Popcorn"
 - 1995 synthetische Sounds bei House und Techno (virtuelle Instrumente)
- 

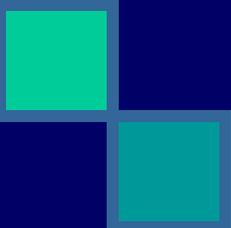


Softwaretechnologien

- 
- **Proprietäre Schnittstellen**
 - **Direct X-Schnittstelle**
 - **Virtuelle Studio Technologie (VST)**
- 

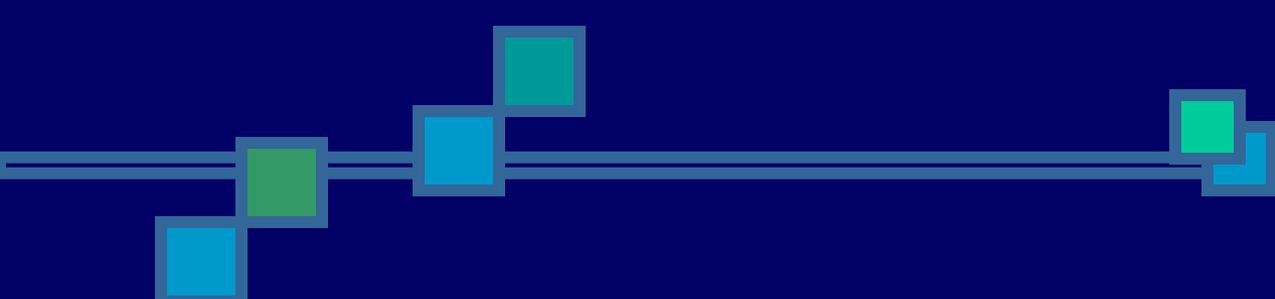


Softwaretechnologien

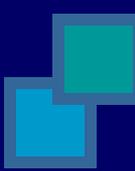


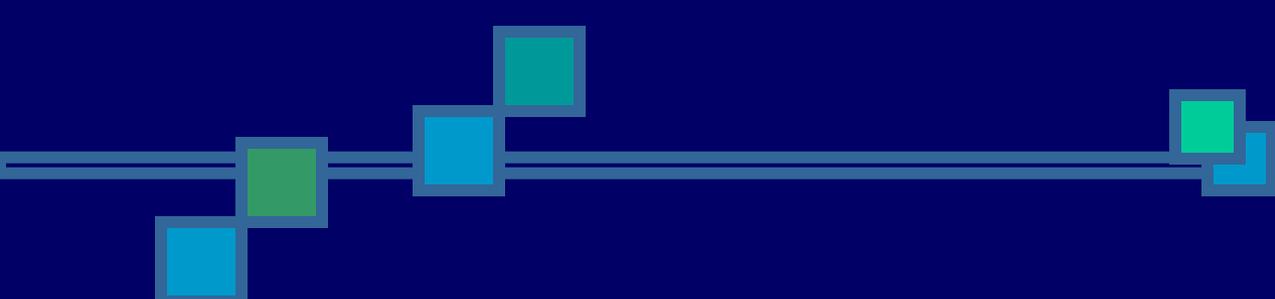
■ Begriff Plugin

- Ein Plugin ist ein Ergänzungs- oder Zusatzmodul (Computerprogramm), das in ein anderes Softwareprodukt „eingeklinkt“ wird. (z.B. Acrobat Reader Plugin für verschiedene Webbrowser)
 - Kommunikation zwischen Hostprogramm und Plugin über klar definierte Schnittstellen
- 

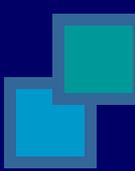


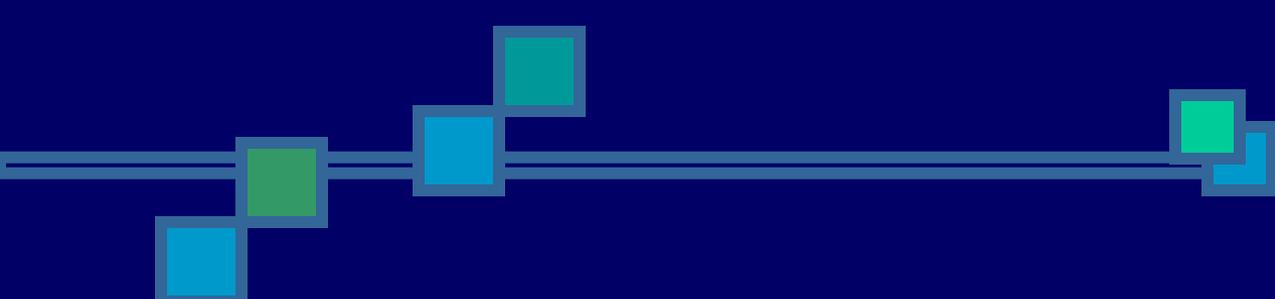
Softwaretechnologien

- **Begriff Plugin bei Audioprogrammen**
 - Hostprogramm ist in der Regel ein Wave-Editor, Harddisk-Recordingprogramm oder Sequenzer
 - Kommunikation über proprietäre Schnittstellen, Direct X oder VST
- 

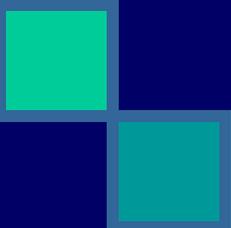
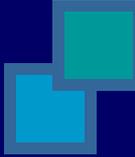


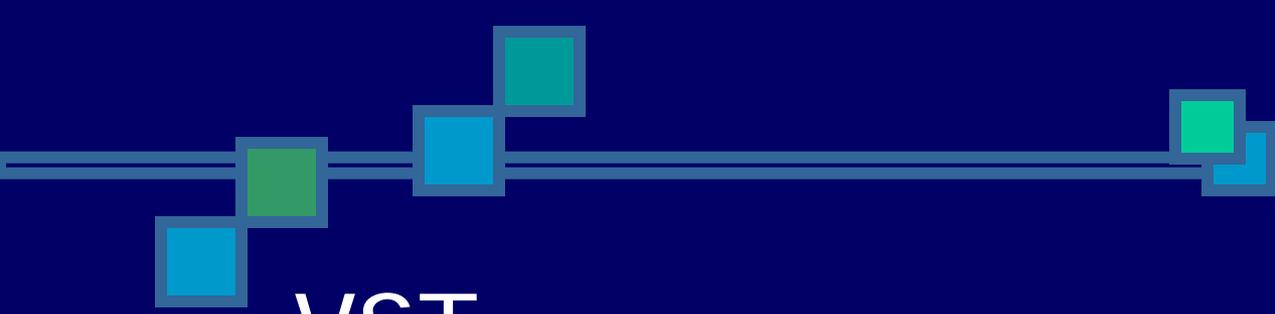
Direct X

- Direct X ist eine Sammlung von APIs für Multimediatechnologien unter Windows
 - Für Musik interessant
 - DirectSound (Wiedergabe und Aufnahme von Soundeffekten)
 - DirectMusic (Wiedergabe z.B. MIDI)
 - DirectShow (Verarbeitung von Video-/Audiodateien)
- 

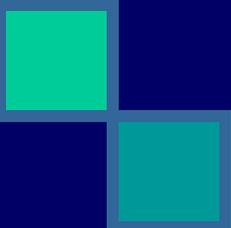


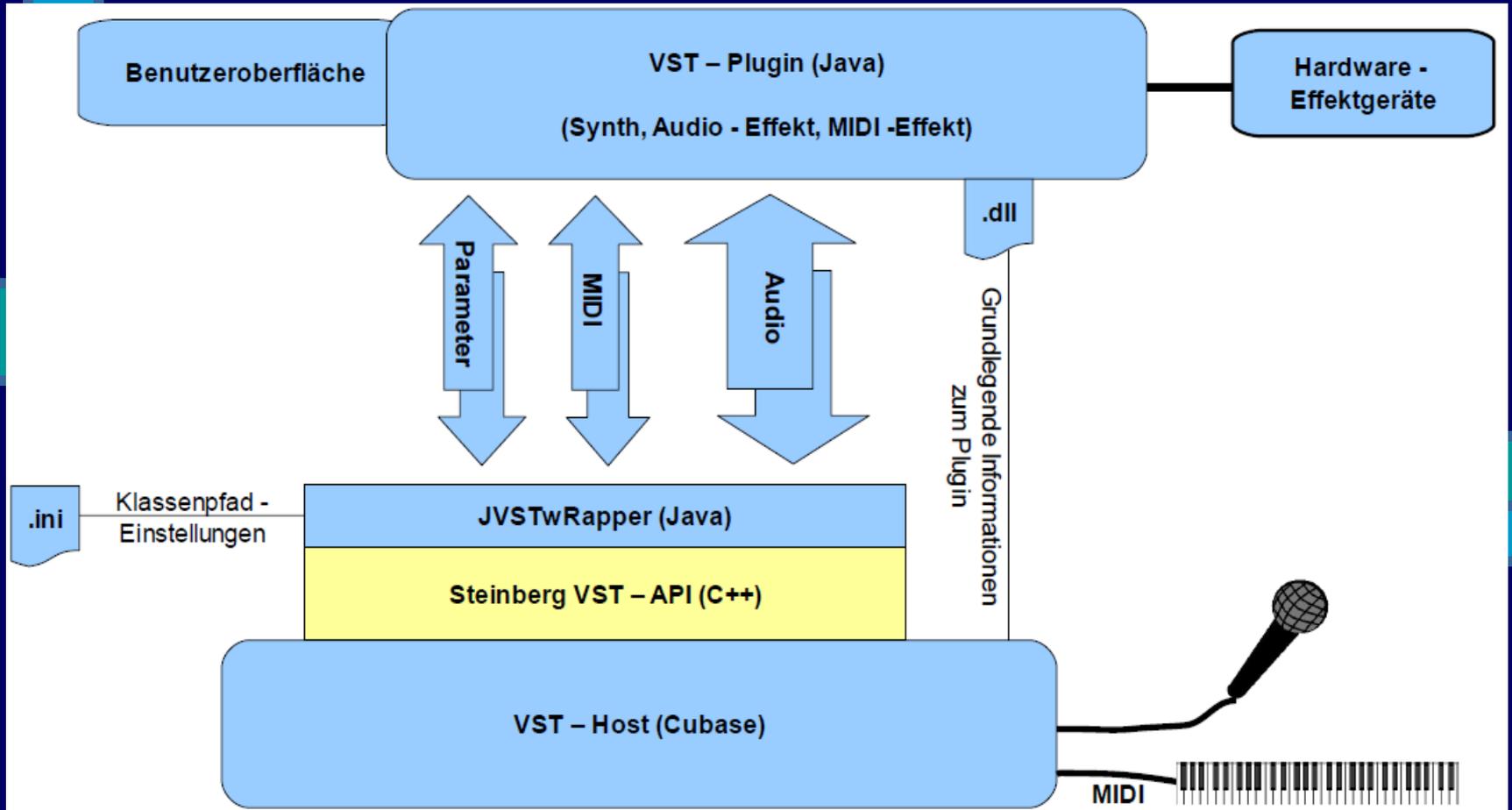
VST

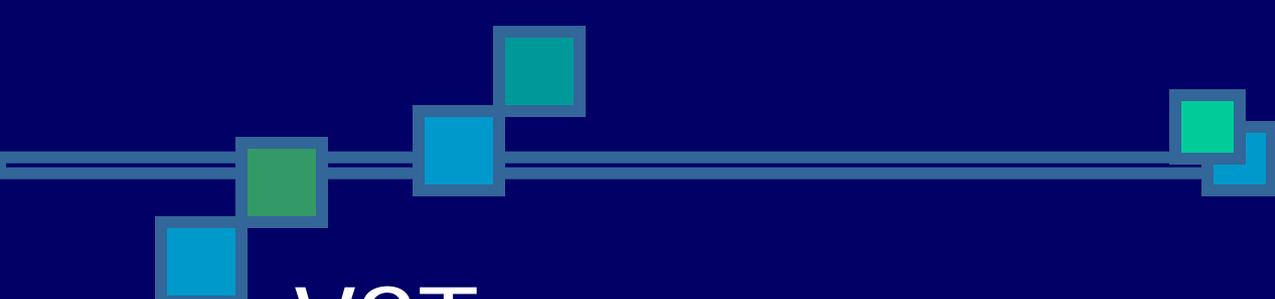
- 
- 1996 Entwicklung der VST-Plugin-Schnittstelle durch Steinberg (für MacOS)
 - 1997 Auch für Windows - Systeme
 - Entwicklung für Dialog zwischen Sequenzer CUBASE als VST-HOST und virtuellen Instrumenten oder Effekte
 - Offenlegung der Schnittstelle für andere Anbieter
 - VST-Instrumente
 - VST-Effekte
- 



VST

- 
- Heute über 5000 Plugins verfügbar
 - Plugin ist unabhängig vom Betriebssystem, es muss ein VST – Gastprogramm installiert sein
 - Für das Gastprogramm ist das Plugin eine .dll – Datei (kann weitere Bibliotheken einbinden)
 - Die VST – Schnittstelle definiert, welche Methoden beiderseits vorhanden sein müssen
- 





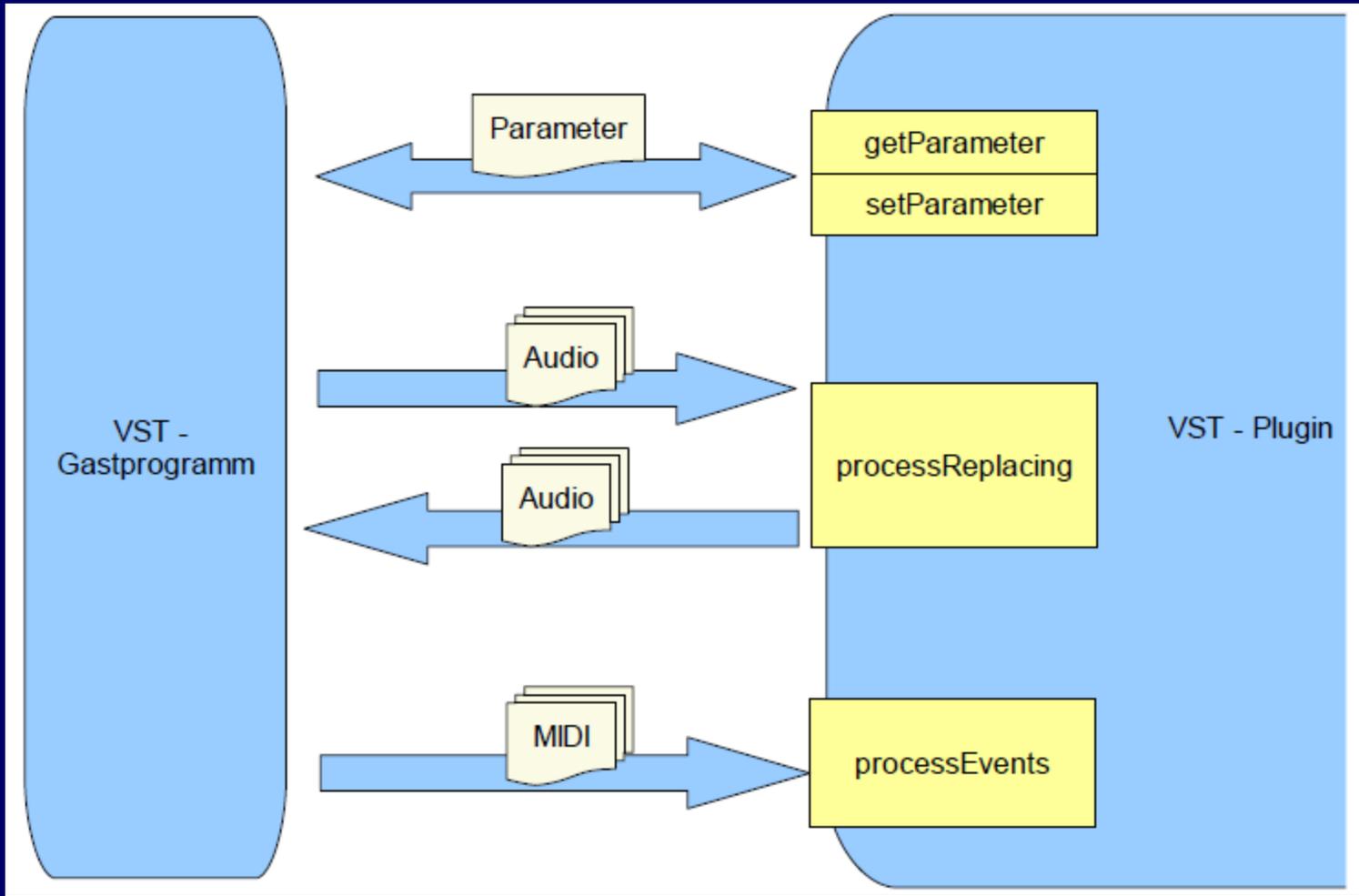
VST

- Das Plugin ist für das Gastprogramm eine „Black Box“
 - Plugin kann eine GUI haben, auf Hardware – Effektgeräte zugreifen oder auch auf mehreren Rechnern im Netzwerk laufen
 - Plugin hat keine Kenntnis davon wo Audiodaten herkommen (Mikrofon, Audiodatei ...)
 - Die VST – Schnittstelle abstrahiert die Datenformate
- 

VST - Datenformate

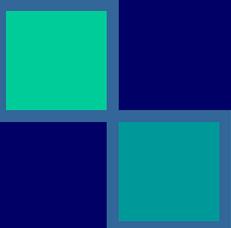
- Audioformat: $-1.0f \dots +1.0f$
Daten werden als Array übergeben
- Parameterformat: $+0.0f \dots +1.0f$
- MIDI – Events als Objekte (VSTMidiEvent)
- Datenaustausch durch Aufruf der Methoden `getParameter()/setParameter()`, `processReplacing()` und `processEvents()`

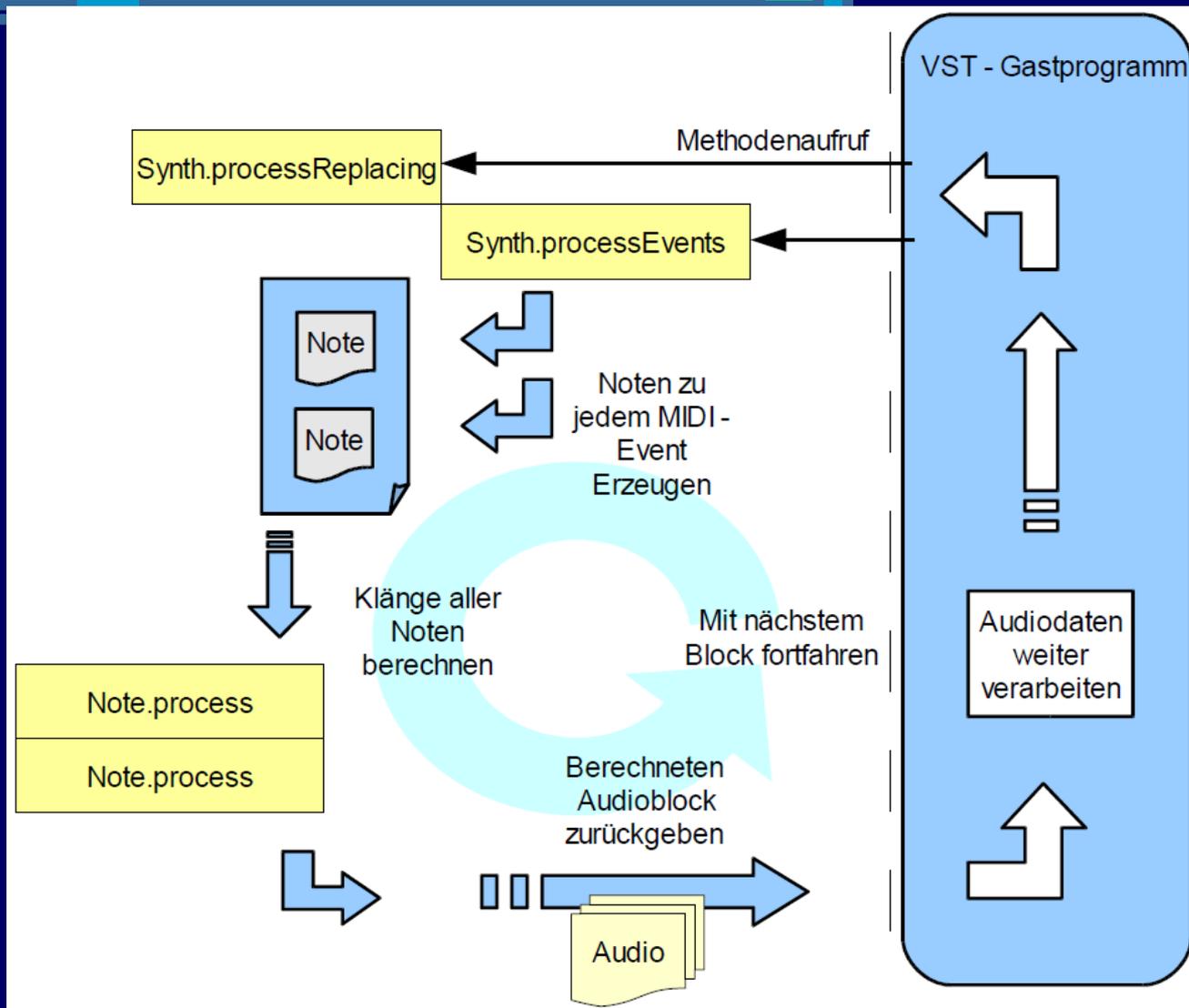
1.0	0 dB
0.5	-6 dB
0.0	$-\infty$ dB

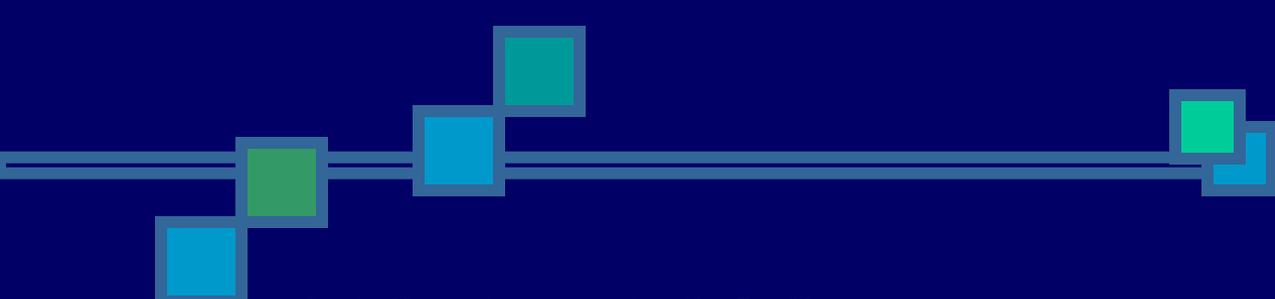




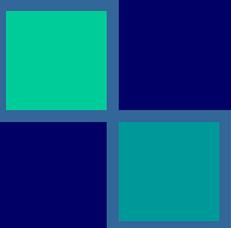
Beispielhafter Programmablauf

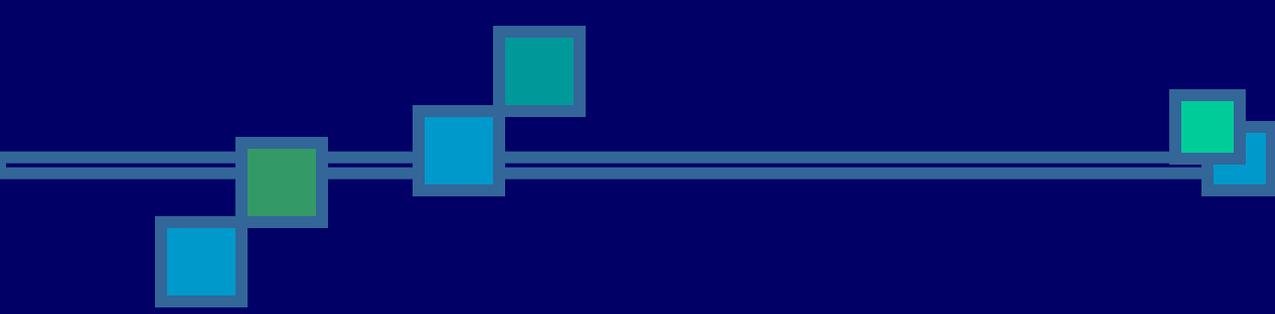
- 
1. Aufruf von `processReplacing()`,
übergabe eines Audioblockes
 2. Aufruf vom `processEvents()`, Übergabe
der MIDI – Daten zum Audioblock
 3. Synthesizer verarbeitet MIDI – Daten
 4. Audioblock zurückgeben
 5. Erneuter Aufruf von `processReplacing()`
- 



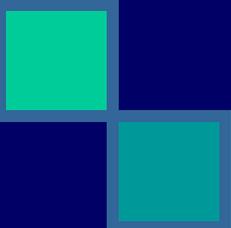
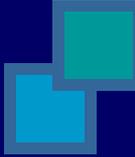


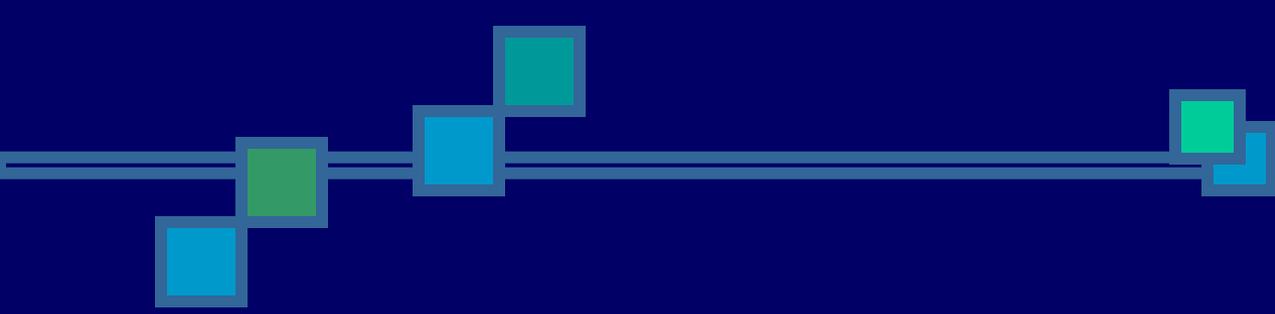
VST - ASIO

- 
- Parallele Entwicklung ASIO-Schnittstelle
 - ASIO = Audio Stream Input/Output
 - Mehrkanalfähiges Audiotransfer-Protokoll
 - Mittels ASIO wird es Software ermöglicht, auf die Multichannel-Fähigkeiten vieler Sound- bzw. Recordingkarten zuzugreifen
 - ASIO ermöglicht geringe Latenzzeiten
- 

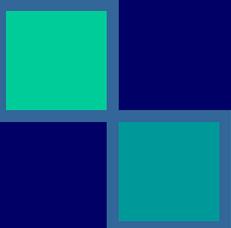
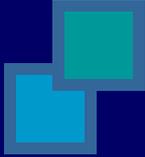


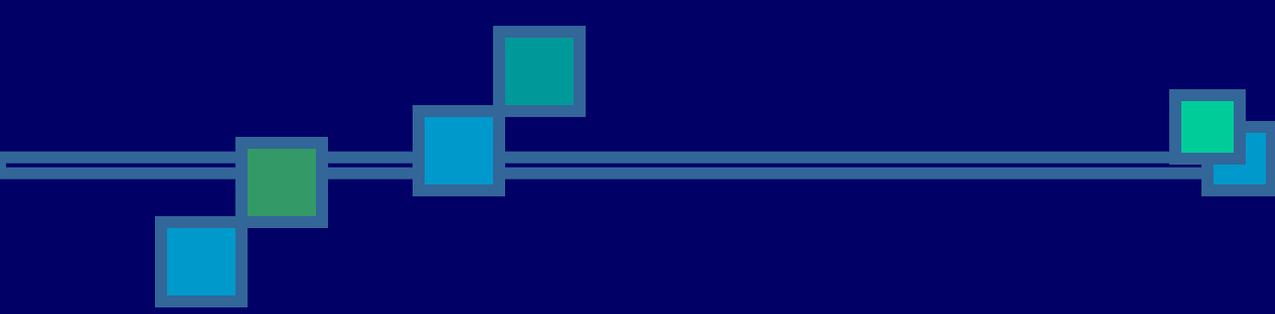
Virtuelle Instrumente

- 
- Synthesizer (z.B. Moog)
 - Orgeln (z.B. Hammond B3)
 - Drums (z.B. Groove Agent)
 - Gitarren (z.B. Virtual Guitarist)
 - Streicher
- 



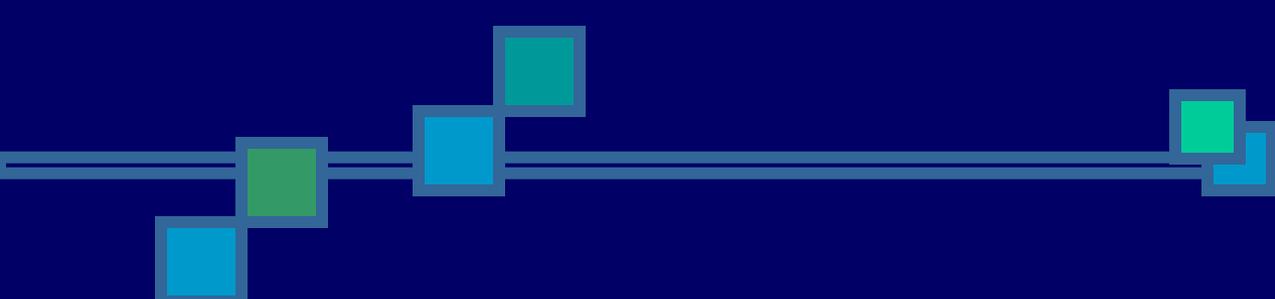
Virtuelle Instrumente

- 
- Demonstration Hammond B3
 - Vortrag Herr Hargarter zu Groove Agent von Steinberg
- 



Virtuelle Effekte

- Chorus
 - Echo, Hall
 - Flanger
 -
 - Demonstration an CUBASIS VST 5.0
(virtuelle Instrumente bzw. Effekte)
- 



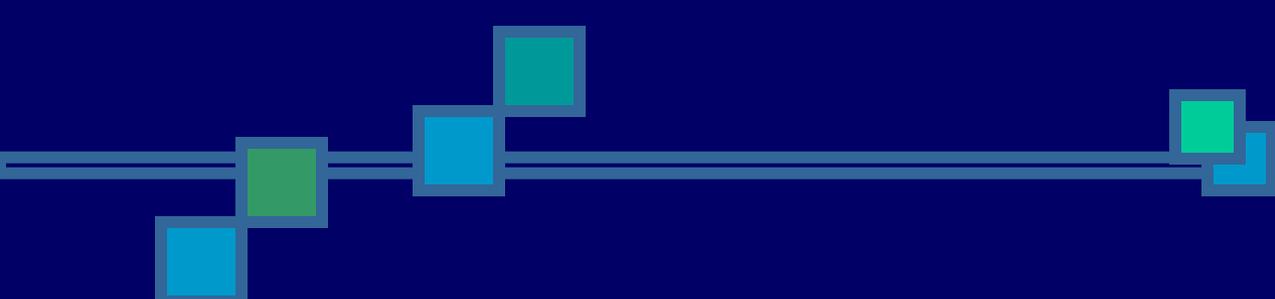
Aufnahmen, Mischen und Mastern

- Praxisteil im Tonstudio durch Tonstudio-Vertreter

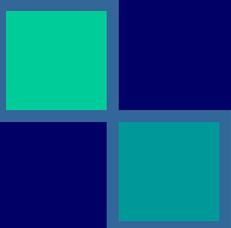
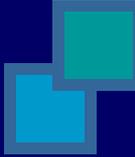
Herrn B. Wittmann (www.blaufabrik.de)

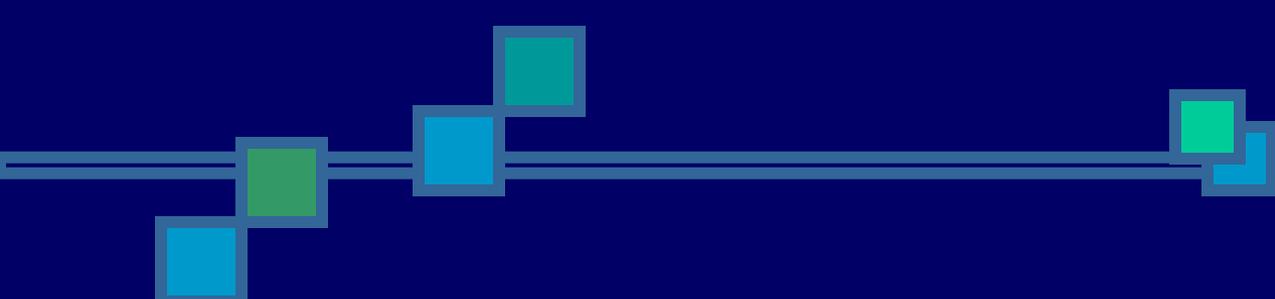
- Beachte Foliensatz zur Aufnahmetechnik auf <http://swlabnt.htw-saarland.de>

Punkt Downloads

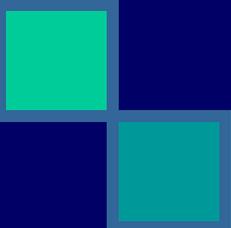


Abschlusspraktikum 2010

- 
- **Hinweise zum Projekt "Gentle Rain" 2010**
 - **Lernziele für Studierende**
 - Kennenlernen Notationsprogramme und Arrangiersoftware
 - Unterschiede MIDI, MP3 und Wav-Format beschreiben und informatiktechnisch erläutern können
 - Generierung und „Komposition“ kleiner Improvisationsteile mittels Arpeggien unter Verwendung von Chromatik
 - Einmischen von generierten Solos im Stile von.... (z.B. Wes Montgomery, Joe Pass)
 - Erzeugung einer kompletten Komposition (Melodieteil, Improvisation, Melodie), Notation, MIDI, MP3 und WAV
- 

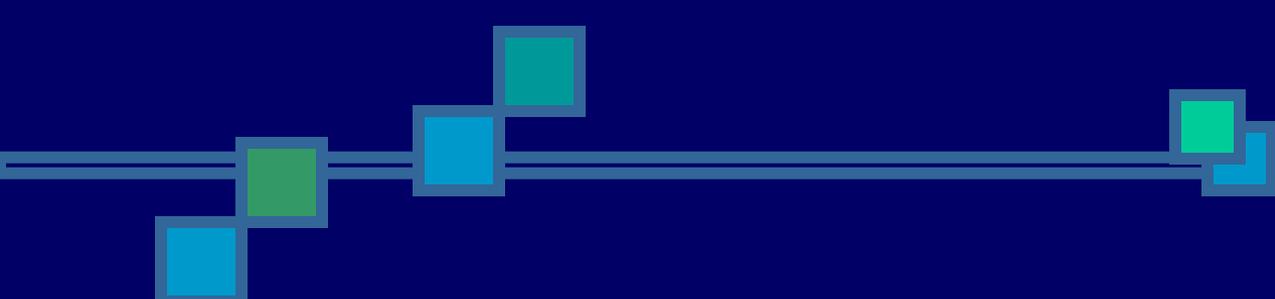


Abschlusspraktikum 2006/2007

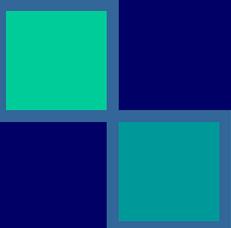


Verwendete Programme

- a. Finale Print Music zur Notation und Erzeugung MIDI-Datei Melodie
 - b. Band in a Box: Import Midi-Datei Melodie, Generierung Begleit-Akkorde, Generierung einer Improvisation, Generierung Midi-Datei Melodie und Begleitung
 - c. Direct Midi to MP3 Converter: Generierung MP3 und Wav-Datei aus Midi-Datei unter Punkt b
- 



Abschlusspraktikum 2008/2009

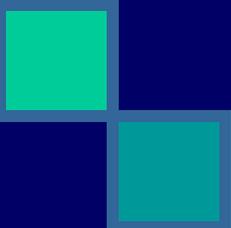


Aufenthalt im Tonstudio

- Kennenlernen Studiotekniken (Mikrophonierung, Aufnahmetechniken, elektronische Instrumente, Abmischsoftware etc.)
 - Aus 2. erzeugter Midi-File bearbeiten (Hall, Echo, Einsatz anderer elektronischer Instrumente, Mastern, Brennen CD)
- 



Abschlusspraktikum 2005/2006/2007/2008

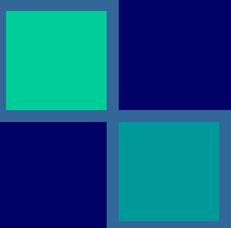


Ergebnisse siehe Webseite
<http://swlabnt.htw-saarland.de>
Download als MP3-Datei



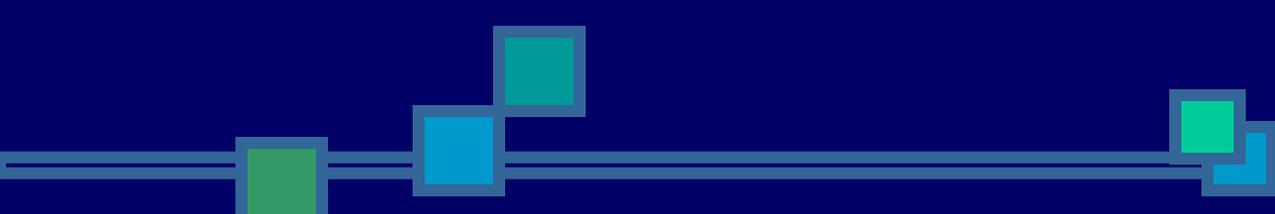


Schnittstellen Informatik und Musik

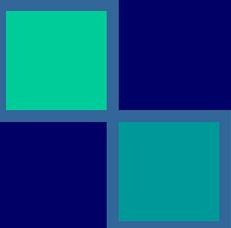


■ Beispiele

- Notation und XML
 - Akkorde und Logik
 - Akkorde und Datenbanken
 - Musikalische Analyse und Pattern Matching
 - Improvisation und „künstliche Intelligenz“
 - Komponieren und Regeln
 - Audio und Kompressionsverfahren
- 

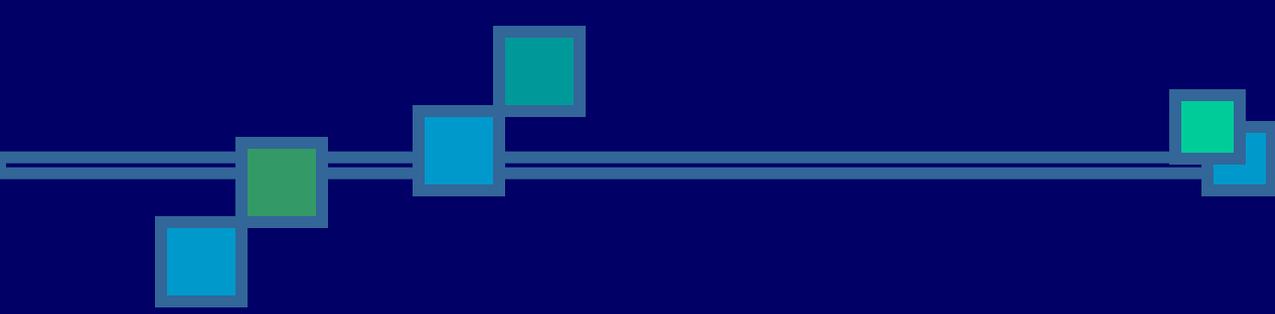


Schnittstellen Informatik und Musik

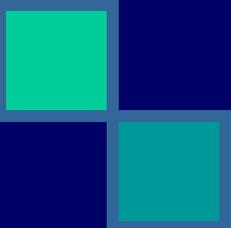


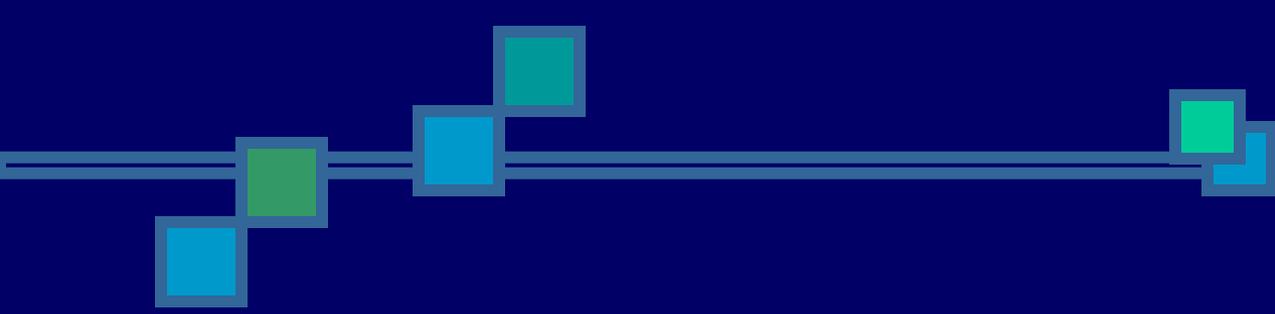
- Beispiele

- MP3 und Fouriertransformationen
 - Musik und Digitalisierung
 - MIDI und XML
 - Instrumente und VST-Technologie
 - Musizieren im Orchester vs. Künstliches Orchester
- 

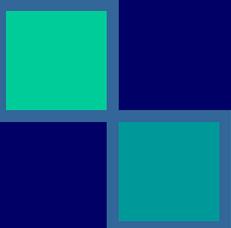


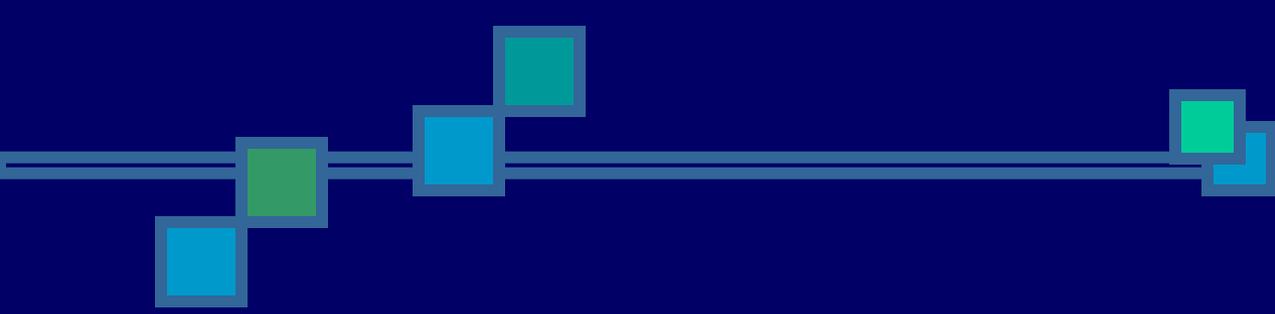
Hauptaussagen Vorlesung

- 
- Computer wird kreativen Prozess des Musizierens noch wesentlicher unterstützen
 - Instrumente und Computer werden verschmelzen
 - Musik- **und** Informatikkenntnisse werden für Musikpädagogen, Hobby- und Profimusiker unerlässlich sein
- 



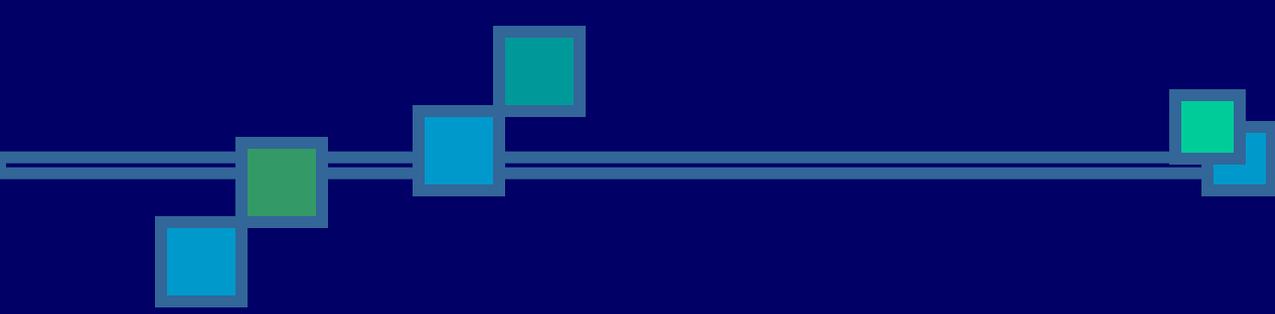
Aussicht

- 
- Entwicklung neuer Software in diesem Bereich setzt fundierte Musik- und Informatikkenntnisse (auch Elektrotechnik) voraus (Musikinformatik)
 - Software muss in der Bedienoberfläche einfacher werden bzw. Tutorials müssen besser werden (integrierte Lernsoftware, nicht nur PDF-Files)
- 

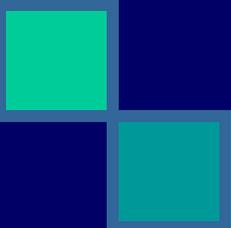


Literatur

- Gorges, Peter: Audio, Midi, MP3. Voggenreiter-Verlag
- Kientzle, Tim: A programmers guide to sound. Addison-Wesley 1998
- Kramarz Volker: Die Pop-Formeln. Voggenreiter 2006
- Malaka, Rainer/Butz, Andreas/Hußmann, Heinrich: Medieninformatik. Pearson Studium 2009
- Mazzola, Guerino: Elemente der Musikinformatik, Birkhäuser 2006
- Neuschwander, H.W.: MIDI – Musik- und MIDI-Grundlagen. Manuskript FH Kaiserslautern
- Nikolaus, U. /Raster, K. : MIDI am PC und Mac. Markt + Technik-Verlag
- Zeitschrift PC und Musik, alle Ausgaben (2010 wurde die Zeitschrift eingestellt)



Internetquellen

- 
- www.musica.at
 - www.midi.org
 - www.musikwissenschaft.uni-mainz.de/Musikinformatik
 - www.ph-karlsruhe.de/~ziegenbalg
(Audiokompression)
 - www.wikipedia.de
- 