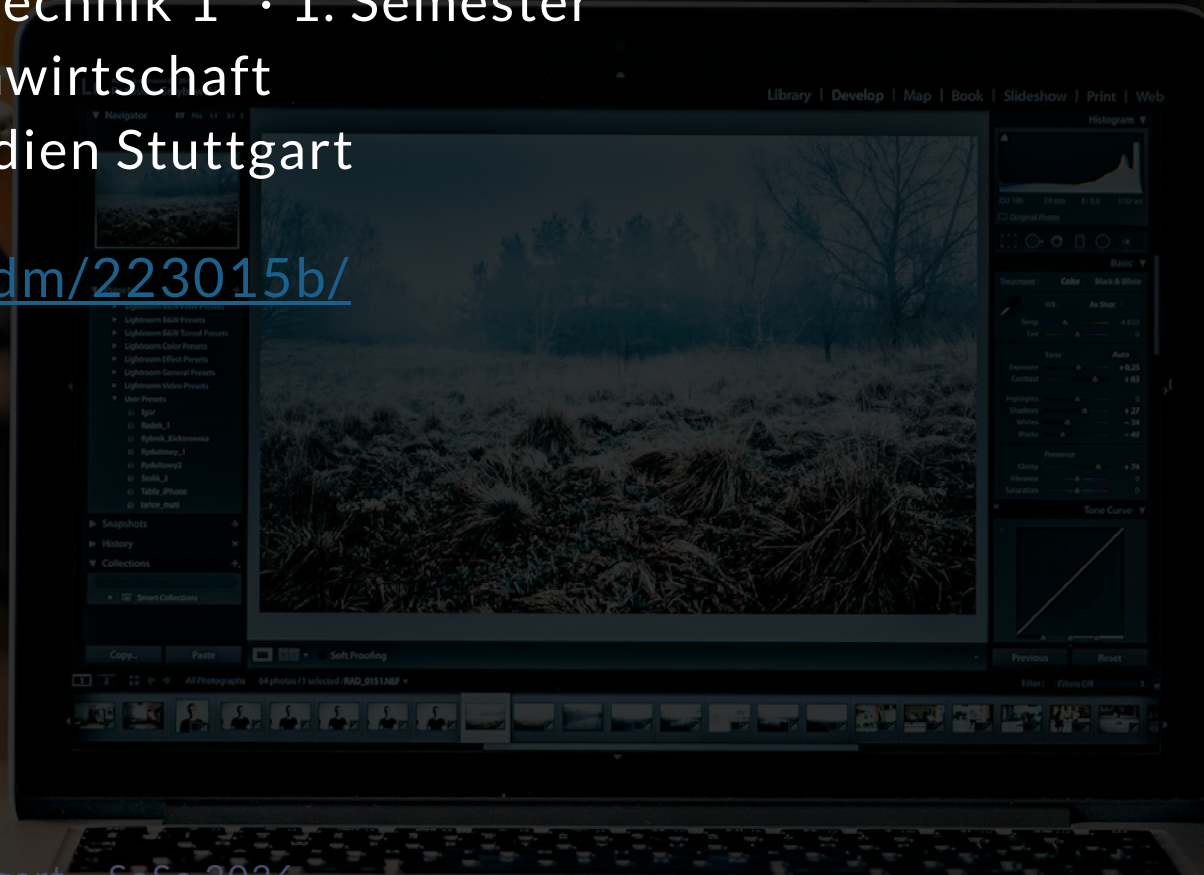


Dateiformate, Schnittstellen, Speichermedien & Distributionswege

223015b · Modul "Technik 1" · 1. Semester
Digital- und Medienwirtschaft
Hochschule der Medien Stuttgart

<https://librete.ch/hdm/223015b/>





Teil 1: Einführung

Grundlagen, Text & Audio

Was sind Daten?

Datenwachstum der Menschheit

Jahr	Datenmenge	Kontext
100.000 v. u. Z.	0	Erste Menschen, nur Sprache
3.000 v. u. Z.	~wenige KB	Keilschrift, Hieroglyphen
1450	~wenige GB	Gutenberg, Buchdruck
1986	2,6 EB	99% analog (Bücher, Vinyl, VHS)
2007	295 EB	94% digital
~2025	~175–181 ZB	Prognose (IDC DataSphere Forecast, 2021)

Der digitale Wendepunkt

Jahr	Analog	Digital	Digital-Anteil
1986	2,6 EB	0,02 EB	1%
2002	—	—	50% (Wendepunkt)
2007	18 EB	277 EB	94%

Perspektive:

- 1986: "Petabyte" war ein theoretisches Konzept
- Magnetband lebt: LTO-Tapes bleiben günstigstes Archivmedium (AWS Glacier, Film-Archive, Rechenzentren)

Digitaler Wendepunkt – Vertiefung

Die Studie von Hilbert & López (Science 332, 2011) analysierte 60 Speichertechnologien von 1986–2007. Der Wendepunkt 2002 markiert den Moment, ab dem mehr Information digital als analog existierte.

Was 1986 „analog“ bedeutete:

- Bücher, Zeitungen, Magazine: ~8 EB
- Vinyl-Schallplatten, Musikkassetten: ~12 EB
- VHS-Kassetten, Filmrollen: ~60 EB

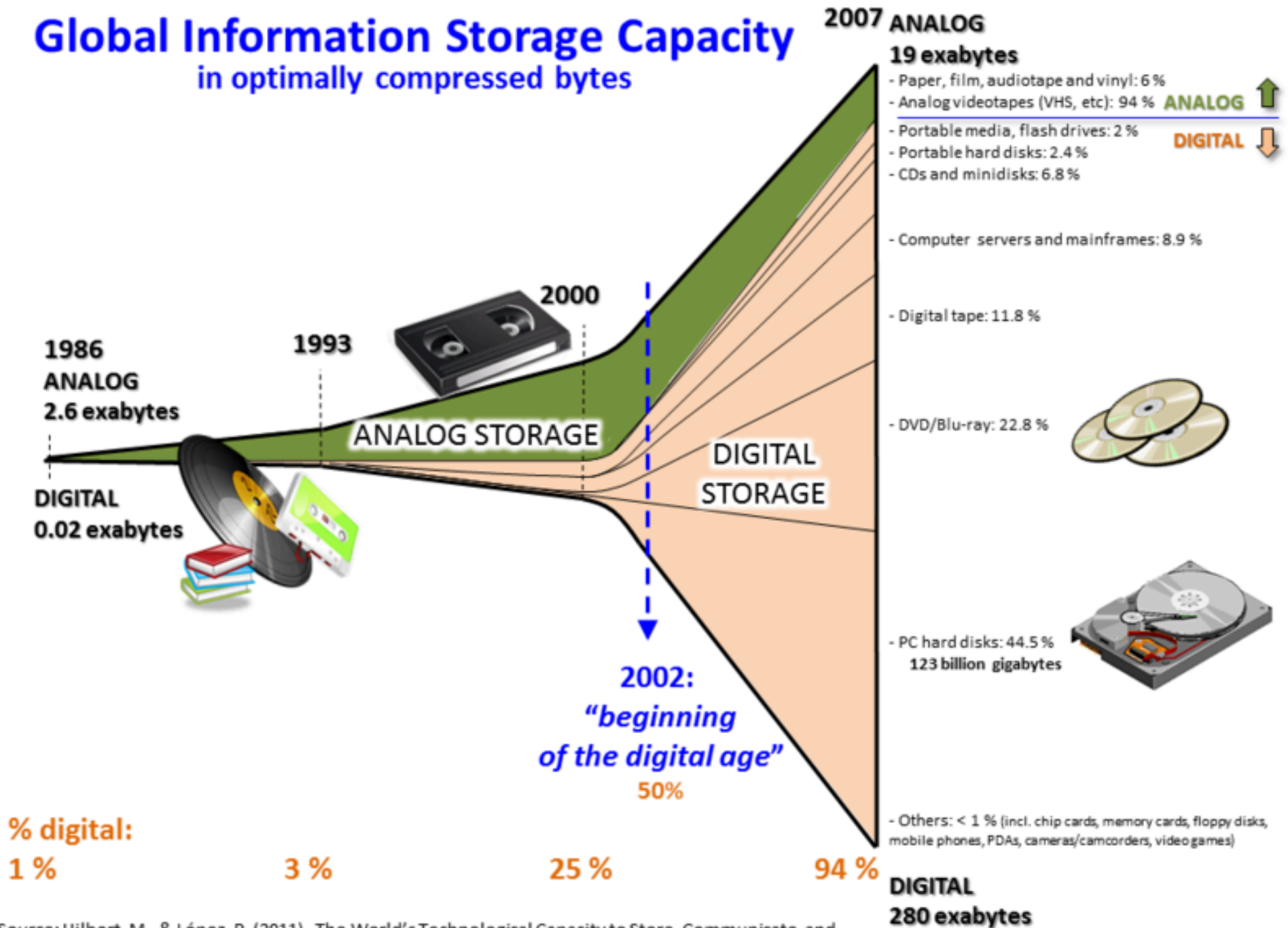
Warum analog stagnierte: Physische Medien haben Kapazitätsgrenzen. Eine VHS speichert ~10 GB; eine Blu-ray (2006) bereits 50 GB auf kleinerer Fläche.

LTO-Magnetband überlebt trotz „alter“ Technologie:

Medium	Kosten/TB	Lebensdauer	Energiebedarf
SSD	~50 €	5–10 Jahre	Dauerstrom
HDD	~15 €	3–5 Jahre aktiv	Dauerstrom
LTO-9	~5 €	30+ Jahre	Nur beim Zugriff

AWS Glacier, Google Coldline und Film-Archive nutzen LTO – langsamer Zugriff, aber unschlagbar günstig und langlebig.

Global Information Storage Capacity in optimally compressed bytes



Source: Hilbert, M., & López, P. (2011). The World's Technological Capacity to Store, Communicate, and Compute Information. *Science*, 332(6025), 60 –65. <http://www.martinhilbert.net/WorldInfoCapacity.html>

181 Zettabyte – Was bedeutet das?

Prognose 2025: Welt erzeugt ca. **175–181 ZB** pro Jahr

- **1 ZB** = 250 Milliarden DVDs
- **29 Terabyte** pro Sekunde (Hochrechnung)

Zum Vergleich:

- 2007: 295 Exabyte (Hilbert & López, Science 2011)
- 2020: ~64 Zettabyte (IDC)
- ~2025: ~175–181 ZB (IDC DataSphere Forecast, Prognose von 2021)

Was sind (digitale) Daten?

- **Text** – Artikel, Gesetzestext, Hausarbeit → `.txt` `.docx` `.pdf`
- **Bild** – Foto, Screenshot, 3D-Rendering, Illustration → `.jpg` `.png` `.svg`
- **Audio** – Song, Podcast, Sprachmemo → `.mp3` `.wav` `.flac`
- **Video** – Reel, Film, Stream → `.mp4` `.mov` `.mkv`

Was sind (digitale) Daten?

- **Text** – Artikel, Gesetzestext, Hausarbeit → `.txt` `.docx` `.pdf`
- **Bild** – Foto, Screenshot, 3D-Rendering, Illustration → `.jpg` `.png` `.svg`
- **Audio** – Song, Podcast, Sprachmemo → `.mp3` `.wav` `.flac`
- **Video** – Reel, Film, Stream → `.mp4` `.mov` `.mkv`

Am Ende alles nur Nullen und Einsen

Bandbreite: Was bedeuten 150 Mbps?

Bandbreite: Was bedeuten 150 Mbps?

150Mbps = 150 Mbit/s

Bandbreite: Was bedeuten 150 Mbps?

150Mbps = 150 Mbit/s \neq ~~150 MB/s~~

Wenn **8 bit (b) = 1 Byte (B)**

dann **150 Mbit/s \div 8 = 18,75 MB/s**

Bandbreite	MB/s	1 GB Film
16 Mbit/s (ADSL)	2 MB/s	~8 Min
50 Mbit/s (ADSL)	6,25 MB/s	~2,7 Min
150 Mbit/s (ADSL)	18,75 MB/s	~54 Sek
1 Gbit/s (Glasfaser)	125 MB/s	~8 Sek

Wie viele Megabyte können bei einer Bandbreite von 250 Mbit/s pro Minute heruntergeladen werden?

- A) 31,25 MB
- B) 1,875 GB = 1.875 MB
- B) 187,5 MB
- C) 250 MB

Upload: Der stille Flaschenhals

ADSL = Asymmetric Digital Subscriber Line

Bandbreite	Download	Upload
16 Mbit/s DSL	16 Mbit/s	1 Mbit/s
50 Mbit/s DSL	50 Mbit/s	10 Mbit/s
150 Mbit/s DSL	150 Mbit/s	40 Mbit/s
Glasfaser (FTTH)	1 Gbit/s	1 Gbit/s

50 MB Upload bei 10 Mbit/s = ~40 Sekunden

Analog vs. Digital

Schall ist eine Druckwelle

- Lautsprecher drückt Luft → Luftmoleküle schwingen
- Die Welle breitet sich aus – kontinuierlich, ohne Stufen
- Vakuum: Keine Schallübertragung möglich (kein Medium)



Schall ist eine Druckwelle

- Lautsprecher drückt Luft → Luftmoleküle schwingen
- Die Welle breitet sich aus – kontinuierlich, ohne Stufen
- Vakuum: Keine Schallübertragung möglich (kein Medium)

Vinyl: Die Rille *ist* die Schallwelle – physische Kopie der Druckschwankung in Rillengeometrie

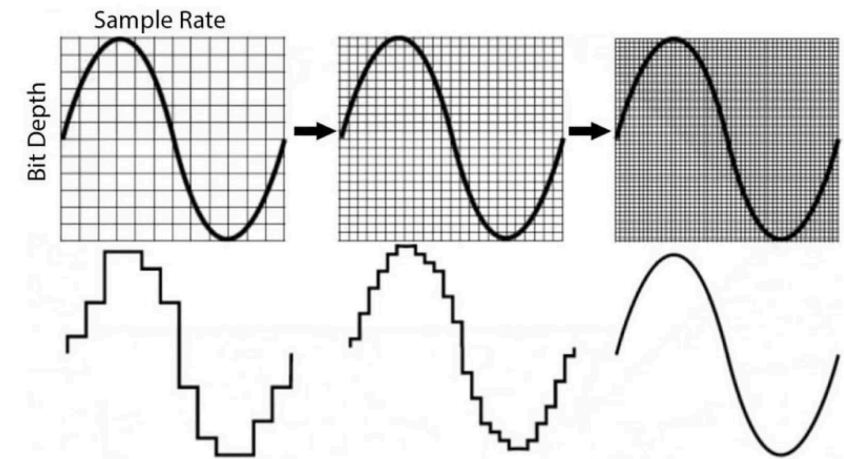


Abtastung (Sampling)

Problem: Eine kontinuierliche Welle lässt sich nicht direkt als Zahl speichern

Lösung: Wir messen sie in regelmäßigen Abständen

- 44.100 Messungen *pro Sekunde* = **Abtastrate** (Sample Rate)
- Einheit: Hz (Hertz) = *pro Sekunde*
- Nyquist-Theorem: min. 2× höchste darzustellende Frequenz
→ $44.100 \div 2 = 22.050$ Hz
(Mensch hört ~20–20 kHz)



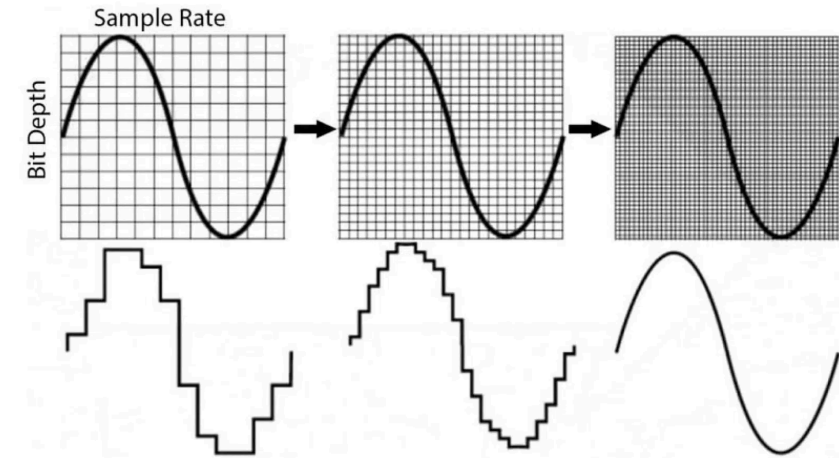
Das digitale Ergebnis

Jede Messung = ein Zahlenwert

Wie genau ist jede Messung? → **Bittiefe (Bit Depth)**

Bittiefe	Stufen	Dynamikumfang
8 Bit	256	~48 dB
16 Bit (CD)	65.536	~96 dB
24 Bit (Studio)	16.777.216	~144 dB

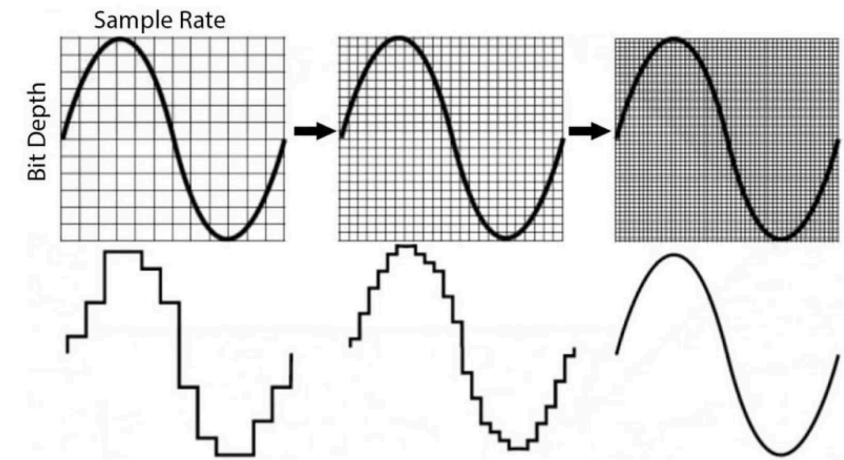
Ergebnis: Eine Datei aus Zahlen



Zurück zu Analog

Beim Abspielen läuft der Prozess rückwärts:

- Zahlen → **DAC** (Digital-Analog-Converter)
- DAC → elektrische Spannung
- Spannung → Lautsprechermembran schwingt
- Membran → Luft schwingt
- Luft → Druckwelle → Ohr

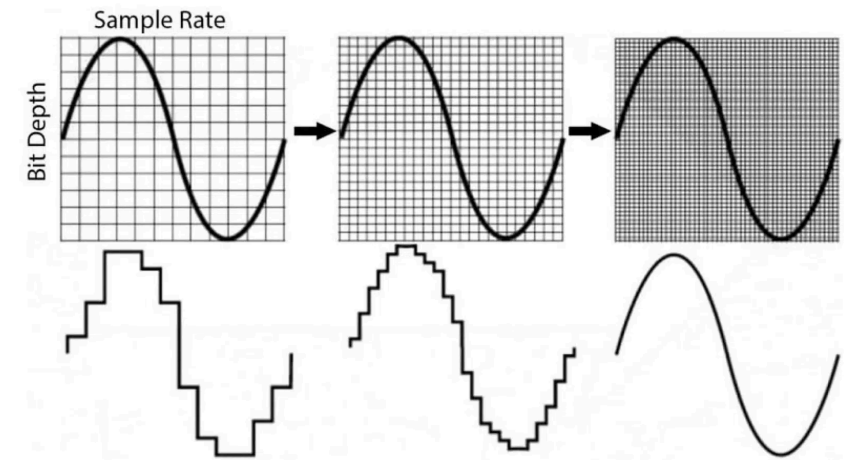


Zurück zu Analog

Beim Abspielen läuft der Prozess rückwärts:

- Zahlen → **DAC** (Digital-Analog-Converter)
- DAC → elektrische Spannung
- Spannung → Lautsprechermembran schwingt
- Membran → Luft schwingt
- Luft → Druckwelle → Ohr

Was wir hören ist wieder analog



Das Problem der Datengröße

Das Problem der Datengröße oder knappen Speichers

COMPACT
d **DISC**
DIGITAL AUDIO

Ein konkretes Beispiel

Eine Minute Musik in CD-Qualität:

44.100 Messungen (pro Sekunde)



Ein konkretes Beispiel

Eine Minute Musik in CD-Qualität:

44.100 Messungen (pro Sekunde)
× **16 Bit** (pro Messung)



Ein konkretes Beispiel

Eine Minute Musik in CD-Qualität:

44.100 Messungen (pro Sekunde)

× **16 Bit** (pro Messung)

× **2 Kanäle** (Stereo)



Ein konkretes Beispiel

Eine Minute Musik in CD-Qualität:

44.100 Messungen (pro Sekunde)

× **16 Bit** (pro Messung)

× **2 Kanäle** (Stereo)

× **60 Sekunden**

= **10,6 MB pro Minute**



Ein konkretes Beispiel

Eine Minute Musik in CD-Qualität:

44.100 Messungen (pro Sekunde)

× **16 Bit** (pro Messung)

× **2 Kanäle** (Stereo)

× **60 Sekunden**

= **10,6 MB pro Minute**

= **10.600.000 Byte pro Minute**



Ein konkretes Beispiel

Eine Minute Musik in CD-Qualität:

44.100 Messungen (pro Sekunde)

× **16 Bit** (pro Messung)

× **2 Kanäle** (Stereo)

× **60 Sekunden**

= **10,6 MB pro Minute**

= **10.600.000 Byte pro Minute**

= **84.800.000 bit pro Minute**



Das Problem skaliert

Inhalt	Unkomprimiert
1 Song (4 Min)	~42 MB
1 Album (60 Min)	~635 MB
10.000 Songs	~420 GB

Kontext 1990er:

- Übliche Festplattengröße: 100–500 MB
- 56 kbit/s Modem → Einfache Songs laden Stunden



Video eskaliert

Eine Minute 4K-Video (unkomprimiert):

3840 × 2160 Pixel (Auflösung pro Bild)

× **3 Byte** (pro Pixel – RGB)

× **30 Bilder** (pro Sekunde)

× **60 Sekunden**

= **~45 GB pro Minute**

Kinofilm in üblicher Länge (~120min) über **5 TB**
(Terabyte)



Artemis II orbitiert

	Apollo (1969)	Artemis II (2026)
Bandbreite	~50 kbit/s	260 Mbit/s
Video	SW, ~320 Zeilen	HD live, 4K gespeichert
Codec	analog	H.265 (HEVC)

260.000.000 Bit/s

÷ 8 (Bit pro Byte) = 32,5 MB/s

× 60 Sekunden = **1,95 GB pro Minute**

Apollo: 50 kbit/s = 0,375 MB/min



Kompressionsraten in der Praxis

Medium	Unkomprimiert	Komprimiert	Faktor
1 Song (4 Min)	~42 MB	~4 MB (MP3 128)	~10×
1 Foto (12 MP)	~36 MB	~3 MB (JPEG)	~12×
1 Min 4K-Video	~45 GB	~375 MB (H.264)	~120×

Zwei Arten der Datenkompression

Zwei Arten von Datenkompressionsalgorithmen

Verlustfreie Kompression (Lossless)

Prinzip: Redundanz entfernen

Verlustfreie Kompression (Lossless)

Prinzip: Redundanz entfernen

Beispiel **Laufängerkodierung** (Run-Length-Encoding – RLE):

```
Original:   AAAAABBBCCCCCCC (16 Zeichen)  
Komprimiert: 5A3B8C      (6 Zeichen)
```

→ Komprimiert 62% kleiner und **100% wiederherstellbar**

Verlustfreie Kompression (Lossless)

Prinzip: Redundanz entfernen

Beispiel **Laufängenkodierung** (Run-Length-Encoding – RLE):

```
Original:   AAAAABBBCCCCCCC (16 Zeichen)  
Komprimiert: 5A3B8C          (6 Zeichen)
```

→ Komprimiert 62% kleiner und **100% wiederherstellbar**

Anwendung:

- ZIP-Archive, PNG-Bilder, FLAC-Audiodateien, RAW-Dateien, Programmcode

Verlustbehaftete Kompression (Lossy)

Prinzip: Irrelevanz entfernen → *Wozu Daten speichern, die niemand wahrnimmt?*

Verlustbehaftete Kompression (Lossy)

Prinzip: Irrelevanz entfernen → *Wozu Daten speichern, die niemand wahrnimmt?*

Originaldatei nicht wiederherstellbar (!)

Die Frage: Was nimmt ein Mensch eigentlich (nicht oder sehr schlecht) wahr?

- Das Ohr hört nicht alle Frequenzen (hohe und tiefe Töne) gleich gut
- Laute Töne überdecken leise Töne (Maskierung)
- Das Auge sieht nicht alle Farbnuancen gleich scharf
- Luminanz (Helligkeit/Dunkelheit) besser erkennen als Chrominanz (Farbunterschiede/Farbigkeit)
- und Vieles mehr

Verlustfrei vs. Verlustbehaftet

	Verlustfrei (Lossless)	Verlustbehaftet (Lossy)
Prinzip	Redundanz entfernen	Irrelevanz entfernen
Reversibel	Ja	Nein (Information unwiederbringlich verloren)
Reduktion	30–50%	80–99%
Formate	ZIP, PNG, FLAC, GIF, ...	JPEG, MP3, H.264/H.265, ...

Faustregel:

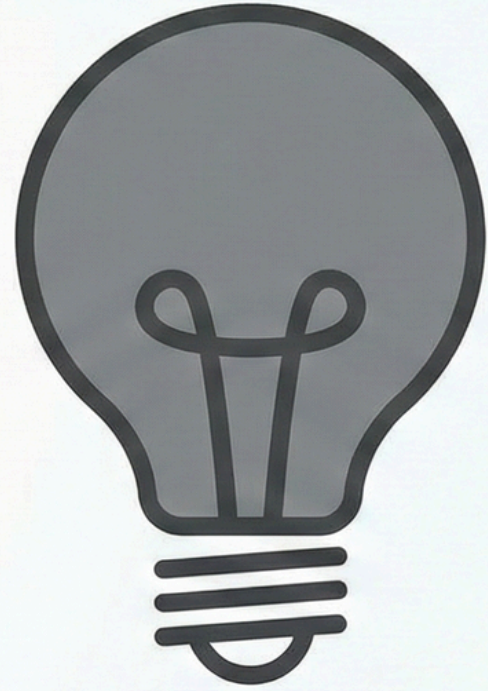
- Medien für EndnutzerInnen (Ton, Bild, Film) → Lossy oft akzeptabel
- Quellmaterial, Code, Archive (RAW) → Lossless nötig

Die Grundbausteine

Bits, Bytes und ihre Darstellung



1



0



Das bit

Kleinste *logische* Informationseinheit

- **0 oder 1**
- AN oder AUS
- Strom fließt oder nicht
- Schwarz oder Weiß
- Richtig oder Falsch
- Take the red pill or blue pill

Das Byte

Die kleinste *adressierbare* Informationseinheit

Das Byte

Die kleinste *adressierbare* Informationseinheit

8 bit = 1 Byte

0 0 1 0 1 0 1 0

= 42

Das Byte

Die kleinste *adressierbare* Informationseinheit

8 bit = 1 Byte

0 0 0 0 0 0 0 0	= 0
0 0 0 0 0 0 0 1	= 1
0 0 0 0 0 0 1 0	= 2
0 0 0 0 0 0 1 1	= 3
0 0 0 0 0 1 0 0	= 4
0 1 0 0 0 0 0 0	= 64
0 1 1 1 1 1 1 1	= 127
1 1 1 1 1 1 1 1	= ?

Das Byte

Die kleinste *adressierbare* Informationseinheit

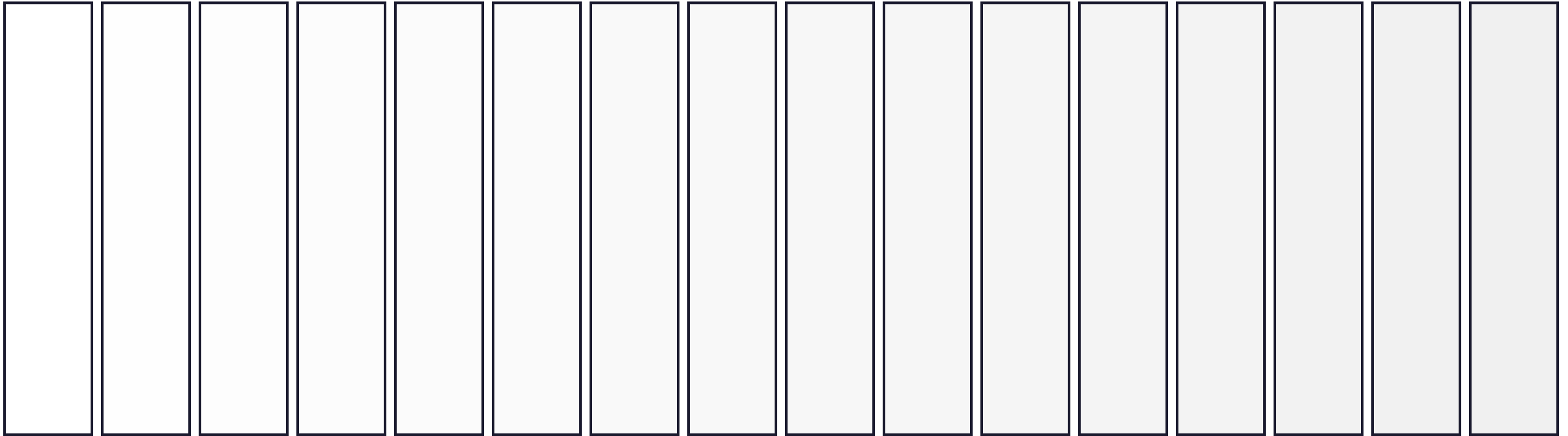
8 bit = 1 Byte

0 0 0 0 0 0 0 0	= 0
0 0 0 0 0 0 0 1	= 1
0 0 0 0 0 0 1 0	= 2
0 0 0 0 0 0 1 1	= 3
0 0 0 0 0 1 0 0	= 4
0 1 0 0 0 0 0 0	= 64
0 1 1 1 1 1 1 1	= 127
1 1 1 1 1 1 1 1	= ?

$2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 = 2^8 = 256$ mögliche Zustände

»256 Shades of Gray«

Zoom: die 16 hellsten Abstufungen



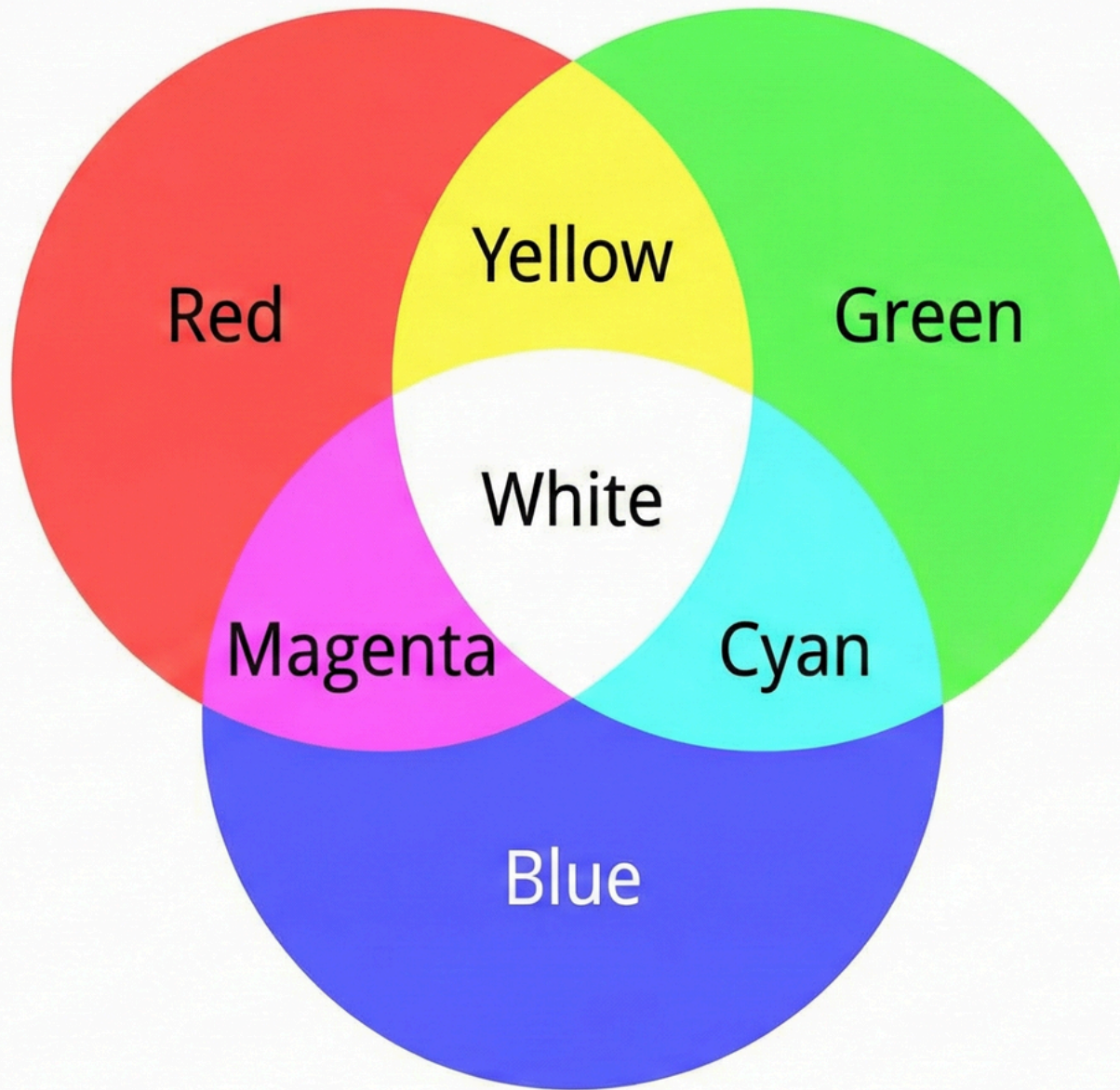
255 254 253 252 251 250 249 248 247 246 245 244 243 242 241 240

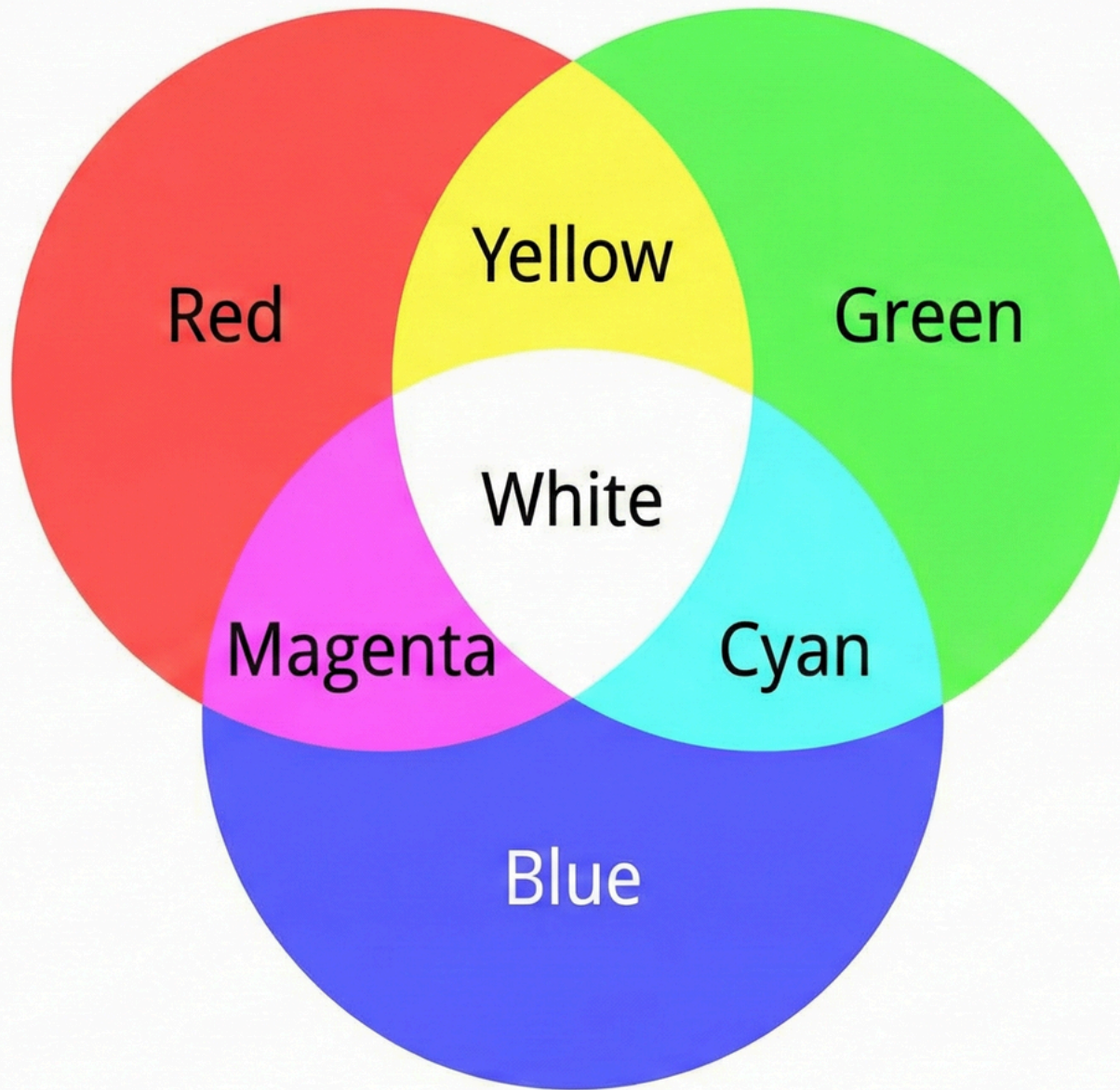
0xFF 0xFE 0xFD 0xFC 0xFB 0xFA 0xF9 0xF8 0xF7 0xF6 0xF5 0xF4 0xF3 0xF2 0xF1 0xF0

jede Stufe = Unterschied von $1/255 \approx 0,4\%$ Helligkeit

Was kann man mit 256 Zuständen machen?

- **Zeichen** (Buchstaben, Zahlen, Symbole)
- **Helligkeit bzw. Luminanz** (0 = Schwarz/Dunkel, 255 = Weiß/Hell)
- **Lautstärke** (dB)
- **Zahlen 0-255** (oder -128 bis +127)





RGB Color Model: Additive Mixing (3 Bytes per Pixel)

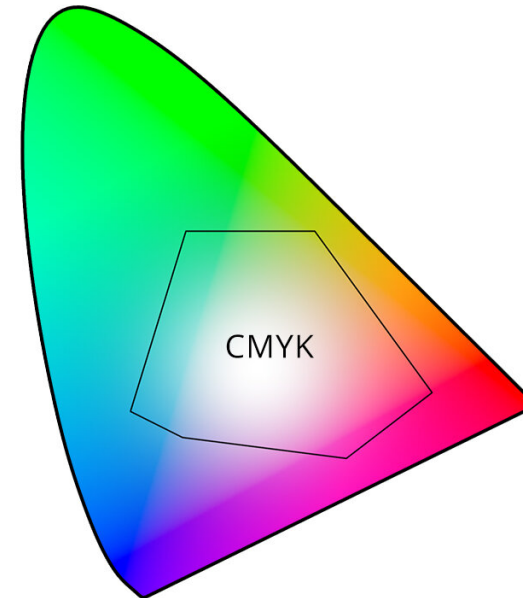
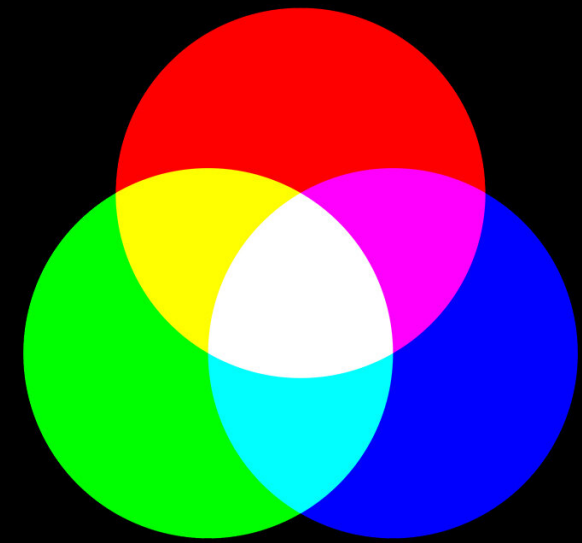
Farben: RGB-Modell

1 Pixel = 3 Byte

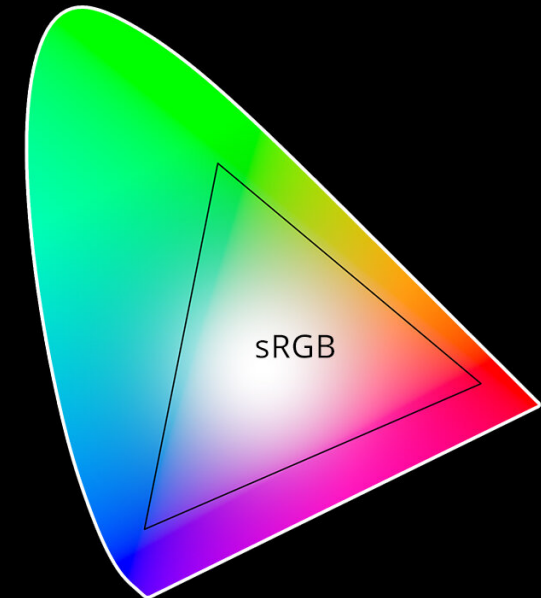
- **Rot:** 0–255
- **Grün:** 0–255
- **Blau:** 0–255

Beispiele:

FF 00 00	= Rot
00 FF 00	= Grün
00 00 FF	= Blau
00 00 00	= Schwarz
FF FF FF	= Weiß



Subtraktiv



Additiv

Das Problem: Sprachen

Die Welt hat mehr als 256 Zeichen (!)

- Englisch Alphabet: 52 (A–Z, a–z)
- ◦ Ziffern: 10 (0–9)
- ◦ Sonderzeichen: ~30

≈ 90 Zeichen passen problemlos in 1 Byte

Jedoch ohne ä, ö, ü, ß, é, à, ç, α, β, , ,  

Unicode: Ein Standard für alle

Unicode (1991): Jedes Schriftsystem der Welt

>150.000 Zeichen:

- Latein, Kyrillisch, Arabisch, Chinesisch, Japanisch...
- Mathematische Symbole, Emoji, historische Schriften

UTF-8: Variable Länge (1–4 Byte pro Zeichen)

- **Zeichen 0–127: identisch mit ASCII** (Abwärtskompatibilität!)
- 1.112.064 gültige Codepunkte
- Umlaute: 2 Byte · CJK: 3 Byte · Emoji: 4 Byte

Beispiel: Byte zählen

Text: "Hello·🌸· ·(Kon-ni-chi-wa)"

Zeichen	Byte
Hello·	$6 \times 1 = 6$ Byte (ASCII)
🌸	4 Byte (Emoji)
·	1 Byte
·	
·	
·	
·	
·(Kon-ni-chi-wa)	16 Byte (ASCII)

Gesamt: 42 Byte für 29 sichtbare Zeichen

1 Byte → 2 Nibble → 2 Hex-Ziffern

Jedes Byte lässt sich sauber halbieren – und 4 Bit passen exakt auf eine Hex-Ziffer.

1 BYTE · 8 BIT · $2^8 = 256$ WERTE



2 NIBBLE · JE 4 BIT · $2^4 = 16$ WERTE



2 HEX-ZIFFERN · SYMBOLE 0-F



01001101 (bin) = **4D** (hex) = **77** (dez) = **"M"** (ASCII)

$16 \times 16 = 256 \cdot 2^4 \times 2^4 = 2^8$ · Darum passt Hex so gut zu Bytes.

Warum 8 Bit? Speicher-Adressierung

CPU adressiert **byteweise** — halbe Byte existieren nicht.

ADRESSE	HEX	BIT (WAS IM SPEICHER STEHT)
0x0000	48	01001000
0x0001	65	01100101
0x0002	6C	01101100
0x0003	6C	01101100
0x0004	6F	01101111
0x0005	21	00100001

→ read 0x0000 ✓

✗ read 0x0000.5 – gibt's nicht

→ read 0x0001 ✓

✗ read 0x0001.5 – gibt's nicht

→ read 0x0002 ✓

→ ...

Byte = kleinste adressierbare Einheit.

Speichercontroller, Bus und CPU sind alle auf 8-Bit-Häppchen ausgelegt.

Einzelne Bit lesen → muss erst `byte = mem[addr]`, dann mit `byte & 0b1000_0000` Bit isolieren.

Hexadezimal

Die Sprache der Datei-Analyse

Hexadezimal: Lesbarkeit

Für den Menschen ungeeignet:

`01010000 01001110 01000111`

Hexadezimal (Base 16):

`50 4E 47`

`5 = 0101` `0 = 0000` = $5 * 16 + 0 = 80$

`4 = 0100` `E = 1110` = $4 * 16 + ? = ?$

`4 = 0100` `7 = 0111` = ?

Jede Hex-Ziffer = 4 Bits (ein "Nibble")

0–9, A–F (10=A, 11=B, ..., 15=F)

Hex → Dezimal

Dezimalwert = (Zeile × 16) + Spalte

×16 / +	0	1	2	3	4	5	6	7	8	9	A (10)	B (11)	C (12)	D (13)	E (14)	F (15)
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
A (10)	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
B (11)	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
C (12)	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
D (13)	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
E (14)	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
F (15)	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

ASCII (0-127) Non-ASCII (128-255)

Beispiel: **4D** → Zeile **4** × 16 + Spalte **D** = 64 + 13 = **77** (= "M" in ASCII)

1 Byte = 8 Bit = 2 Hex = 1 ASCII-Zeichen

Dieselbe Datei — dieselben Byte, drei Schreibweisen. Jeder Rahmen = ein Byte.

DATEI.BIN Roh-Bit · 8 Bit pro Rahmen

01001000	01100101	01101100	01101100	01101111	00101100	00100000	01010111
01101111	01110010	01101100	01100100	00100001	00100000	01001000	01101111
01110111	00100000	01100001	01110010	01100101	00100000	01111001	01101111
01110101	00111111	00001010	... (weitere Byte)				

JE 8 BIT → 2 HEX-ZIFFERN

DATEI.HEX Hex-Sicht · 2 Hex-Ziffern pro Rahmen

48	65	6C	6C	6F	2C	20	57
6F	72	6C	64	21	20	48	6F
77	20	61	72	65	20	79	6F
75	3F	0A	...				

JE 1 BYTE → 1 ZEICHEN

DATEI.TXT Text-Sicht · 1 ASCII-Zeichen pro Rahmen

H	e	l	l	o	,		W
o	r	l	d	!		H	o
w		a	r	e		y	o
u	?	↵	...				

☐ = Leerzeichen (0x20) ☐↵ = Zeilenumbruch (0x0A, nicht druckbar) **Byte ändern sich nicht.** Nur die Anzeige.

Dieselben 8 Byte — drei Perspektiven

PNG-Datei-Anfang: 89 50 4E 47 0D 0A 1A 0A

1 · BITSTREAM
was wirklich gespeichert wird

10001001 01010000 01001110 01000111 00001101 00001010 00011010 00001010



2 · HEX
gruppiert in 8-Bit-Häppchen

89 50 4E 47 0D 0A 1A 0A



3 · BEDEUTUNG
was die Byte sagen

89

MAGIC BYTE

> 127 → "Ich bin Binär, kein Text"

50 4E 47

FORMAT-KÜRZEL

P · N · G (ASCII) → "PNG-Datei"

0D 0A 1A 0A

TRANSPORT-SCHUTZ

CR · LF · EOF · LF → erkennt kaputte Übertragung

Byte ändern sich nicht — nur unsere **Brille**. Bitstream zeigt das *was*, Hex das *wie kompakt*, Bedeutung das *warum*.

Wo begegnet Ihnen Hex-Code?

Kontext	Beispiel
CSS-Farben	#FF5733
MAC-Adressen	00:1A:2B:3C:4D:5E
Fehlercodes	0x80070005
Speicheradressen	0xA04F20
Unicode	U+00E4 (ä)
Datei-Signaturen	89 50 4E 47 (PNG)

Bit vs. Byte: Die Verwirrung

1 Byte = 8 Bit → $\text{Bit} \div 8 = \text{Byte}$

Einheit (Bit)	Einheit (Byte)
1 Kbit = 1.000 Bit	1 KB = 1.000 Byte = 8.000 Bit
1 Mbit = 1.000.000 Bit	1 MB = 1.000.000 Byte = 8 Mbit
1 Gbit = 1 Mrd. Bit	1 GB = 1 Mrd. Byte = 8 Gbit

100 Mbit/s Bandbreite = **12,5 MB/s** Downloadrate

ASCII

One *Zeichensatz* to rule them all

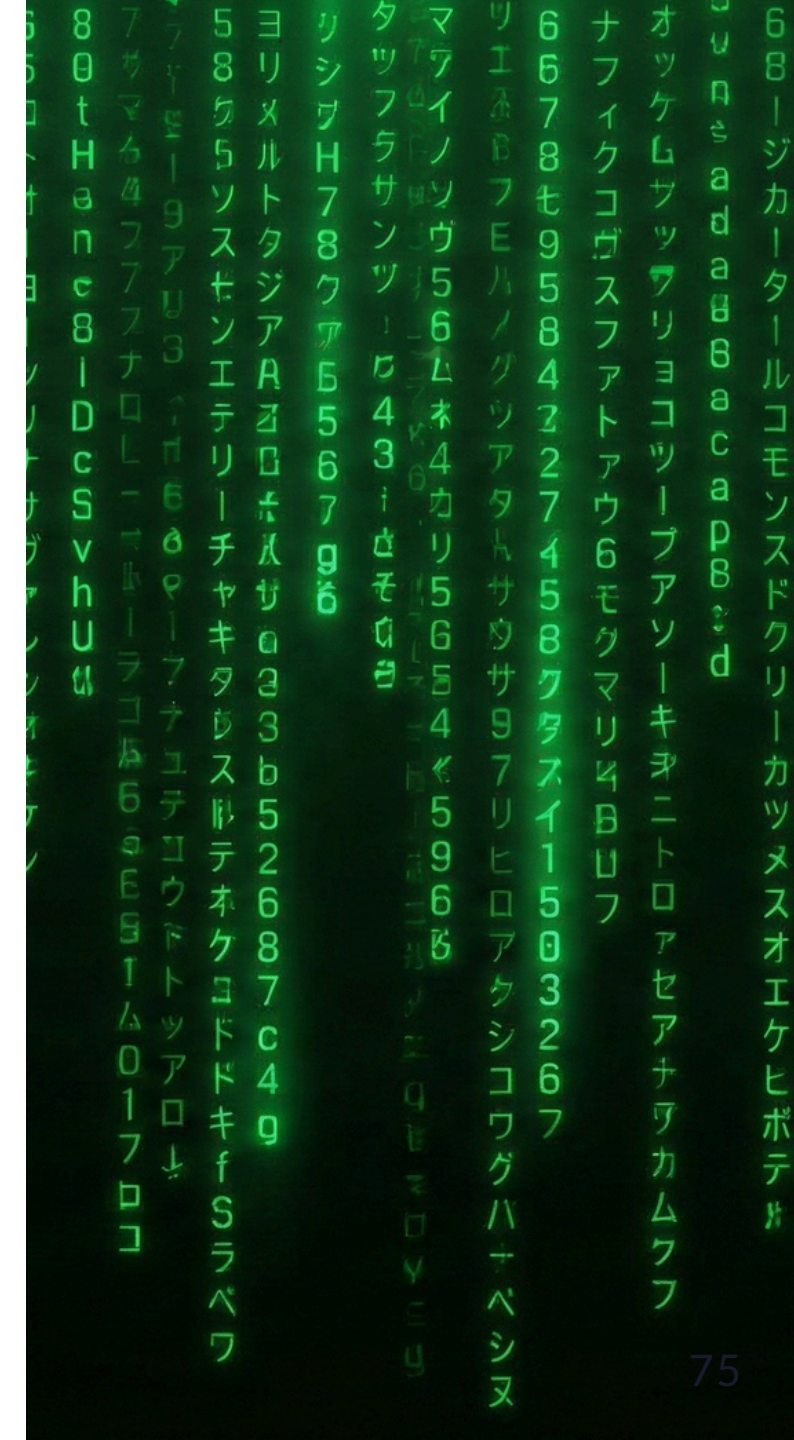
Bits					0	0	0	0	1	1	1	1
					0	0	1	1	0	0	1	1
b ₄	b ₃	b ₂	b ₁	Column	0	1	2	3	4	5	6	7
↓	↓	↓	↓	Row	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	\$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(8	H	X	h	x
1	0	0	1	9	HT	EM)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	
1	1	0	1	13	CR	GS	-	=	M]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	_	o	DEL

What the HEX-Code

89 50 4E 47 ...

Binär	Hex	Dez	ASCII
1000 1001	89	137	× (> 127)
0101 0000	50	80	P
0100 1110	4E	78	N
0100 0111	47	71	G

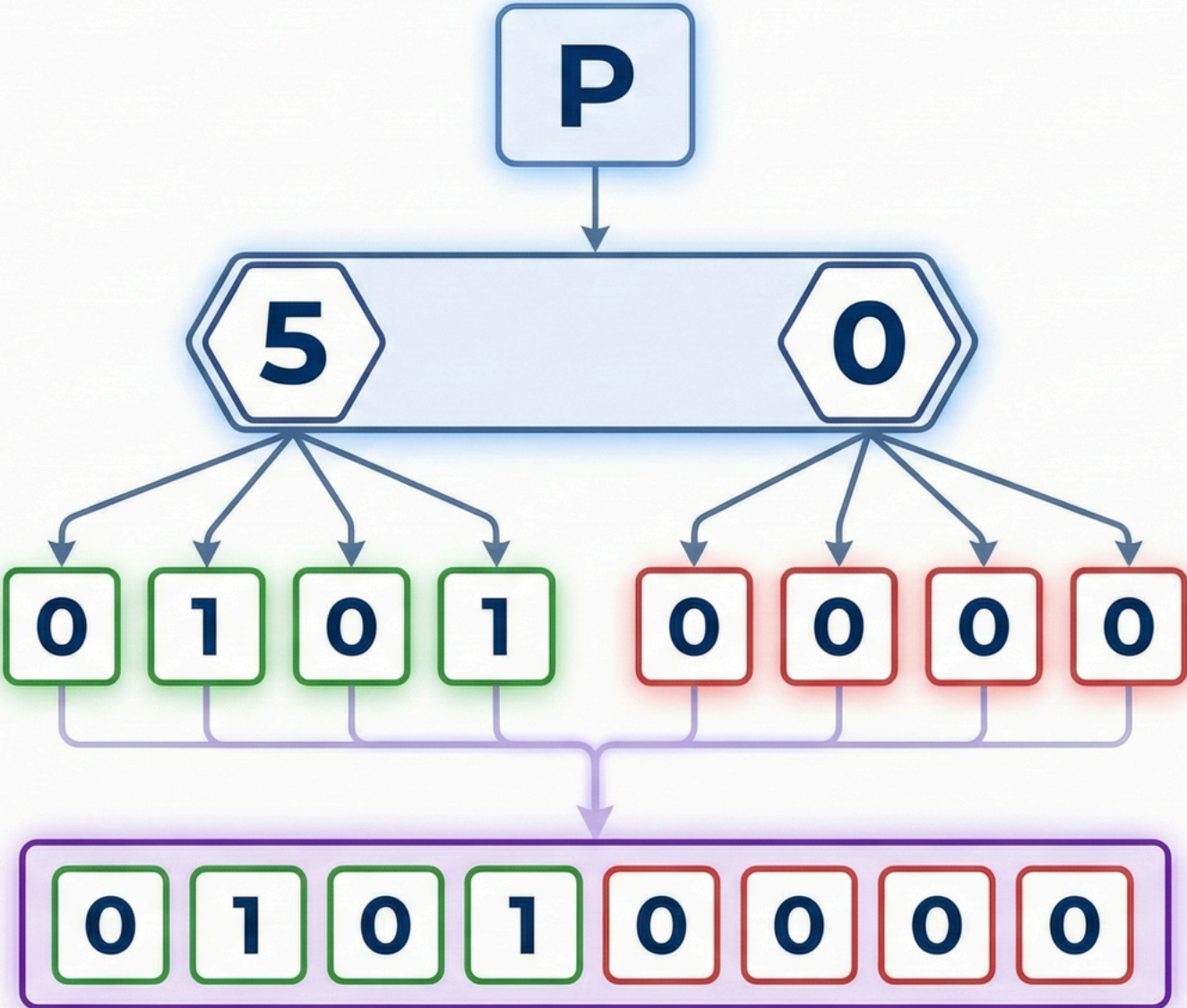
→ **89** übersteigt den ASCII-Raum und markiert eine Binärdatei



00000000:	8950	4e47	0d0a	1a0a	0000	000d	4948	4452
00000010:	0000	0b00	0000	0600	0802	0000	00f4	1859
00000020:	3b00	0003	1669	5458	7458	4d4c	3a63	6f6d
00000030:	2e61	646f	6265	2e78	6d70	0000	0000	003c
00000040:	3f78	7061	636b	6574	2062	6567	696e	3d22
00000050:	efbb	bf22	2069	643d	2257	354d	304d	7043
00000060:	6568	6948	7a72	6553	7a4e	5463	7a6b	6339
00000070:	6422	3f3e	203c	783a	786d	706d	6574	6120
00000080:	786d	6c6e	733a	783d	2261	646f	6265	3a6e
00000090:	733a	6d65	7461	2f22	2078	3a78	6d70	746b
000000a0:	3d22	584d	5020	436f	7265	2035	2e35	2e30
000000b0:	223e	203c	7264	663a	5244	4620	786d	6c6e
000000c0:	733a	7264	663d	2268	7474	703a	2f2f	7777
000000d0:	772e	7733	2e6f	7267	2f31	3939	392f	3032
000000e0:	2f32	322d	7264	662d	7379	6e74	6178	2d6e
000000f0:	7323	223e	203c	7264	663a	4465	7363	7269
00000100:	7074	696f	6e20	7264	663a	6162	6f75	743d
00000110:	2222	2078	6d6c	6e73	3a65	7869	663d	2268
00000120:	7474	703a	2f2f	6e73	2e61	646f	6265	2e63
00000130:	6f6d	2f65	7869	662f	312e	302f	2220	786d

00000000: 8950 4e47 0d0a 1a0a 0000 000d 4948 4452 .PNG.....IHDR
00000010: 0000 0b00 0000 0600 0802 0000 00f4 1859Y
00000020: 3b00 0003 1669 5458 7458 4d4c 3a63 6f6d ;....iTXtXML:com
00000030: 2e61 646f 6265 2e78 6d70 0000 0000 003c .adobe.xmp.....<
00000040: 3f78 7061 636b 6574 2062 6567 696e 3d22 ?xpacket begin="
00000050: efbf bf22 2069 643d 2257 354d 304d 7043 ..." id="W5M0MpC
00000060: 6568 6948 7a72 6553 7a4e 5463 7a6b 6339 ehiHzreSzNTczkc9
00000070: 6422 3f3e 203c 783a 786d 706d 6574 6120 d"?> <x:xmpmeta
00000080: 786d 6c6e 733a 783d 2261 646f 6265 3a6e xmlns:x="adobe:n
00000090: 733a 6d65 7461 2f22 2078 3a78 6d70 746b s:meta/" x:xmptk
000000a0: 3d22 584d 5020 436f 7265 2035 2e35 2e30 ="XMP Core 5.5.0
000000b0: 223e 203c 7264 663a 5244 4620 786d 6c6e "> <rdf:RDF xmln
000000c0: 733a 7264 663d 2268 7474 703a 2f2f 7777 s:rdf="http://ww
000000d0: 772e 7733 2e6f 7267 2f31 3939 392f 3032 w.w3.org/1999/02
000000e0: 2f32 322d 7264 662d 7379 6e74 6178 2d6e /22-rdf-syntax-n
000000f0: 7323 223e 203c 7264 663a 4465 7363 7269 s#"> <rdf:Descri
00000100: 7074 696f 6e20 7264 663a 6162 6f75 743d ption rdf:about=
00000110: 2222 2078 6d6c 6e73 3a65 7869 663d 2268 "" xmlns:exif="h
00000120: 7474 703a 2f2f 6e73 2e61 646f 6265 2e63 ttp://ns.adobe.c
00000130: 6f6d 2f65 7869 662f 312e 302f 2220 786d om/exif/1.0/" xm

ASCII Character 'P' to 8-Bit Binary



Magic Numbers

Dateityp-Identifikation durch erste Bytes

Format	Magic Number (Hex)	Lesbar?
PNG	89 50 4E 47	× P N G
JPEG	FF D8 FF	× × ×
PDF	25 50 44 46	% P D F ✓
ZIP	50 4B 03 04	P K × ×

Wichtig: ASCII = nur 0–127! Werte darüber (z.B. **89** = 137) sind **nicht druckbar** (non-printable). *Hex-Editoren zeigen dafür **.** oder **ü** als Platzhalter.*

Dateneinheiten

Einheit	Bytes	Potenz	Beispiel
Byte	1	10^0	Farbwert eines Pixels
Kilobyte (KB)	1.000	10^3	Kleiner Programmcode
Megabyte (MB)	1 Million	10^6	Textdokument
Gigabyte (GB)	1 Milliarde	10^9	Kinofilm in FullHD
Terabyte (TB)	1 Billion	10^{12}	~12h Video in 4K
Petabyte (PB)	1 Billiarde	10^{15}	Netflix-Gesamtarchiv
Exabyte (EB)	1 Trillion	10^{18}	Alle E-Mails weltweit/Tag
Zettabyte (ZB)	1 Trilliarde	10^{21}	Globale Datenmenge 2025

THE FOURTH REVOLUTION

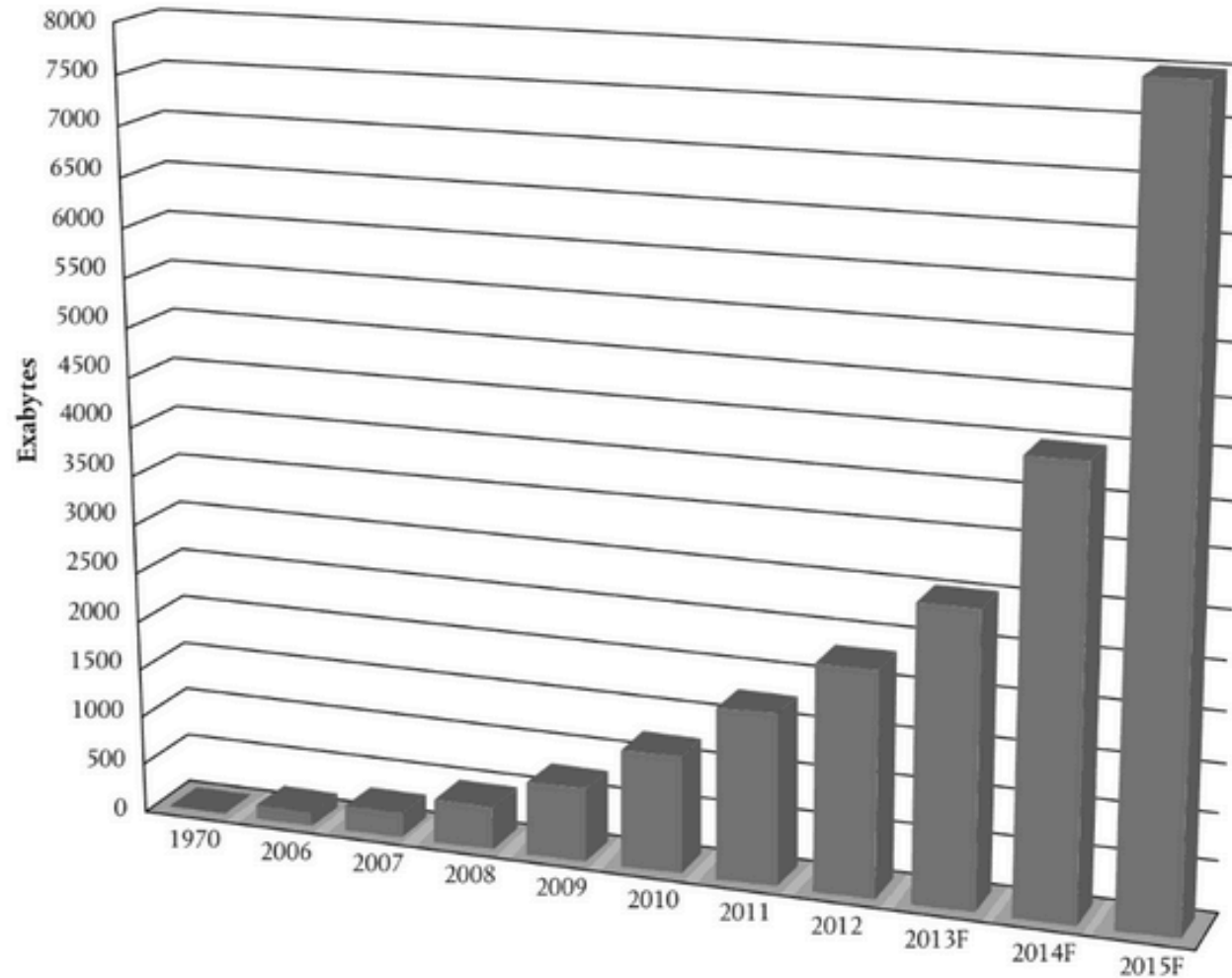


Fig. 10. The Growth of Big Data.

Source: based on IDC white paper, 'The Diverse and Exploding Digital Universe', March 2008, and IDC white paper, 'Worldwide Big Data Technology and Service 2012–2015 Forecast', March 2012.

AI-generierte Inhalte 2025

Wie viel Content ist heute synthetisch?

Bereich	AI-Anteil
Neue Webseiten	~74% enthalten AI-Content
Web-Text gesamt	~30-40% AI-generiert
Neue Artikel	~52% von AI geschrieben
Social-Media-Bilder	~71% AI-generiert

Prognose 2026: 90% des Online-Contents synthetisch

Teil 2: Die MP3-Revolution

Psychoakustik & Audio-Kompression



Analoge Medien

Distribution: physisch (Kauf, Verleih, Kopie)

- **Text**
 - Bücher, Zeitungen, Zeitschriften, Lochkarten
- **Bild**
 - Fotografie (Negativ, Dia, Polaroid), Mikrofilm
- **Audio:**
 - Schallplatte (Vinyl, Schellack), Tonband, Musikkassette
- **Video:**
 - Film (35mm, Super 8), VHS, Betamax

Analoge Medien – Vertiefung

Analoge Speicherung codiert Information als **kontinuierliche physikalische Größe**: Rillentiefe (Vinyl), Magnetfeldstärke (Tonband), Silberkorn-Dichte (Film). Es gibt keine diskreten Stufen – theoretisch unendliche Auflösung, praktisch begrenzt durch Rauschen.

Generationsverlust entsteht, weil jede Kopie neues Rauschen addiert:

- Schallplatte → Kassette: Frequenzgang leidet, Rauschen steigt
- VHS → VHS: Farbsättigung sinkt, Schärfe nimmt ab
- 3. Generation: oft unbrauchbar

Medium	Typische Auflösung	Dynamik
Vinyl (audiophil)	~20–20.000 Hz	~70 dB
Tonband (Studio)	~30–15.000 Hz	~55 dB
35mm Film	~4K-äquivalent	~13 Blendenstufen

Paradox der Analogtechnik: Das Original ist einzigartig und unersetzlich – aber genau deshalb anfällig. Jedes Abspielen einer Schallplatte trägt mikroskopisch Material ab; jeder Filmdurchlauf riskiert Kratzer.

Analoge Medien: Vor- und Nachteile

Vorteile	Nachteile
Kein Abspielgerät nötig (Buch, Foto)	Qualitätsverlust bei jeder Kopie
Haptisches Erlebnis	Physischer Verschleiß
Unabhängig von Strom/Internet	Begrenzte Haltbarkeit
Keine Formatkonvertierung	Platzbedarf bei Lagerung
Eindeutiges Original	Aufwendige Durchsuchbarkeit

Von Analog zu Digital: Die Kopier-Revolution

Das Problem analoger Kopien:

Kassette → Kassette → Kassette = immer schlechter

Was Digital anders macht:

- **Identische Kopien** – kein Qualitätsverlust, nie
- **Einfache Massenproduktion** – Copy & Paste
- **Perfekte Archivierung** – Bits verändern sich nicht

Daher: "Raubkopien"

Der Begriff entstand, weil digitale Kopien *tatsächlich identisch* mit dem Original waren – nicht wie bei Kassetten eine schlechtere Version.

Quelle: c64-wiki.de/wiki/Raubkopie

Digitale Medien

Distribution: Datenträger (CD, USB), Download, Streaming, P2P

- **Text**
 - E-Book (PDF, EPUB), Dokumente (TXT, DOCX)
- **Bild**
 - Digitalfoto (JPEG, PNG, RAW, WebP, GIF)
- **Audio**
 - Audiodatei (MP3, FLAC, WAV, AAC, OGG)
- **Video**
 - Videodatei (MP4, MKV, AVI, WebM)

Digitale Medien – Vertiefung

Digitale Speicherung quantisiert kontinuierliche Signale in diskrete Werte. Der **Quantisierungsfehler** (Differenz zum Original) ist der Preis der Digitalisierung – aber einmal digitalisiert, bleibt die Information exakt.

Bit-identische Kopien revolutionierten die Medienindustrie:

- Keine Qualitätskette mehr: 1000. Kopie = 1. Kopie = Original
- Kosten pro Kopie: praktisch null (nur Speicherplatz)
- Perfekte Archivierung: Bits altern nicht (nur der Datenträger)

Aspekt	Analog	Digital
Kopiervorgang	Physikalischer Prozess	Bit-Kopie
Qualität pro Generation	Verschlechtert	Identisch
Fehlerkorrektur	Unmöglich	Möglich (ECC, RAID)
Formatmigration	Verlust	Verlustfrei möglich

Die Kehrseite: Digitale Obsoleszenz. Ein DOCX von 2025 ist in 50 Jahren womöglich unlesbar – während ein Buch von 1525 heute noch lesbar ist. Offene Formate (PDF/A, FLAC, PNG) mildern dieses Risiko.

Digitale Speichermedien

- **Optische Speicher**
 - CD, DVD, Blu-ray
- **Magnetische Speicher**
 - Festplatte (HDD), Magnetband (LTO)
- **Flash-Speicher**
 - SSD, USB-Stick, SD-Karte
- **Cloud-Speicher**
 - Dropbox, Google Drive, iCloud, AWS S3

Digitale Speichermedien – Vertiefung

Jede Technologie hat physikalische Vor- und Nachteile:

Optisch (CD/DVD/Blu-ray): Laser liest Pits (Vertiefungen) und Lands (Erhöhungen). Robust gegen Magnetfelder, aber empfindlich gegenüber Kratzern und UV-Licht. M-DISC verspricht 1000 Jahre – unter Laborbedingungen.

Magnetisch (HDD/LTO): Magnetisierte Bereiche auf rotierenden Platten oder Band. HDDs haben bewegliche Teile (Verschleiß); LTO-Bänder sind passiv und extrem langlebig, aber sequentieller Zugriff.

Flash (SSD/USB/SD): Elektronen in Floating Gates speichern Bits. Keine beweglichen Teile, aber begrenzte Schreibzyklen (TLC: ~3.000, SLC: ~100.000). Ohne Strom verlieren Zellen nach Jahren ihre Ladung.

Szenario	Empfehlung	Grund
Betriebssystem	NVMe SSD	Geschwindigkeit
Videoarchiv	HDD	Kapazität/Preis
Langzeitarchiv	LTO + M-DISC	Lebensdauer
Austausch	USB/SD	Portabilität

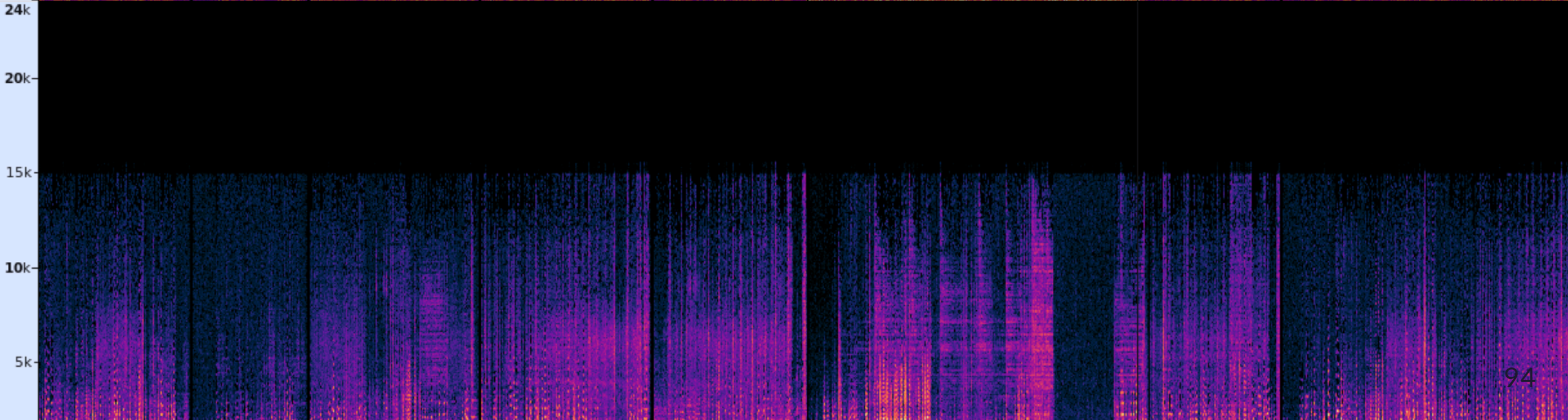
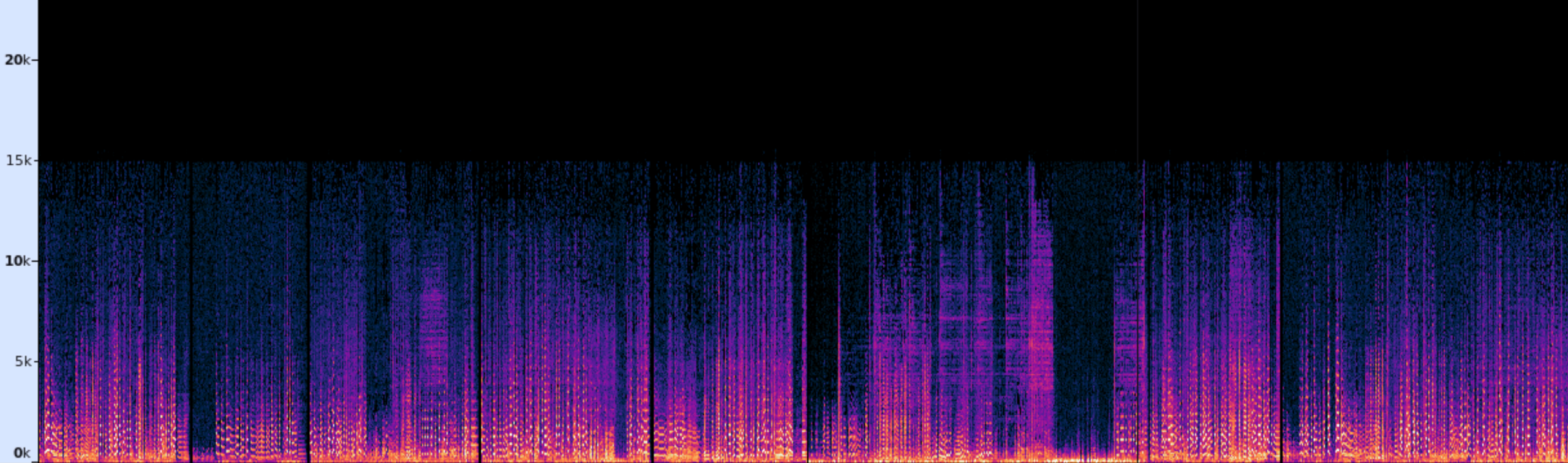
Cloud ist physisch HDD/SSD/LTO in Rechenzentren – kein eigenes Medium, sondern Zugriffsmethode.

Das Speicherproblem der Digitalisierung

Ziel: Analoge Schallwelle möglichst originalgetreu rekonstruieren

CD-Qualität (1982): $44.100 \text{ Hz} \times 16 \text{ Bit} \times 2 \text{ Kanäle} = 10,584 \text{ MB/Minute}$

Inhalt	Größe	Problem (1990er)
1 Song (4 Min)	~42 MB	Ausreichend Speicher
1 Album (60 Min)	~635 MB	Gesamte Festplatte



Die Abtastrate (Sample Rate)

Analog → **Digital** $\hat{=}$ **Kontinuierlich** → **Diskret**

Nyquist-Theorem:

“ Um eine Frequenz zu rekonstruieren, braucht man mindestens **2× so viele Samples**.
44.100 Hz ÷ 2 = **22.050 Hz** max. darstellbare Frequenz
(Mensch hört: ~20 Hz bis ~20.000 Hz → passt!) ”

Die Bittiefe (Bit Depth)

Wie genau messen wir jeden Punkt?

Bittiefe	Stufen	Dynamikumfang
8 Bit	256	~48 dB
16 Bit (CD)	65.536	~96 dB
24 Bit (Studio)	16.777.216	~144 dB

16 Bit = 2^{16} = 65.536 Lautstärkestufen
(von absoluter Stille bis maximaler Lautstärke)

Abtastrate (Sample Rate) × Bittiefe (Bit Depth)

Zwei Dimensionen der Digitalisierung:

Dimension	Was bedeutet es?	CD-Qualität
Abtastrate (Sample Rate)	Messungen pro Sekunde (horizontal)	44.100 Hz
Bittiefe (Bit Depth)	Genauigkeit pro Messung (vertikal)	16 Bit

44.100 Hz × 16 Bit × 2 Kanäle = 10,584 MB/Minute

Kompression

Weniger Daten, gleiche(?) Information

Wo liegt der Hebel für Kompression?

CD-Qualität: $44.100 \text{ Hz} \times 16 \text{ Bit} \times 2 \text{ Kanäle} = \mathbf{10,6 \text{ MB/Min}}$

MP3 (128 kbps): = **~1 MB/Min** (Faktor 10!)

Container-Parameter (das Raster):

Parameter	Reduzieren →	Konsequenz
Abtastrate	Weniger Messpunkte/Sek	Max. Frequenz sinkt
Bittiefe	Weniger Lautstärkestufen	Mehr Rauschen
Kanäle	Mono statt Stereo	Kein Raumklang

Psychoakustik: Der MP3-Trick

Inhalt (was durchs Raster geht):

Methode	Reduzieren →	Konsequenz
Psychoakustik	Unhörbare Frequenzen	Kaum wahrnehmbar

- **MP3 nutzt hauptsächlich Psychoakustik**
- Container bleibt ähnlich, Inhalt wird "ausgedünnt"

Die Geburt der MP3

1982: Universität Erlangen-Nürnberg
Karlheinz Brandenburg, Diplom-Ingenieur

1987: Fraunhofer IIS entwickelt MPEG-1 Audio Layer III

1988: Patentanmeldung

1992: Erste Software-Implementierung

1995: .mp3 Dateiendung offiziell

Karlheinz Brandenburg

"Vater der MP3"

- Diplom-Ingenieur, Universität Erlangen-Nürnberg
- Fraunhofer IIS (Institut für Integrierte Schaltungen)
- Forschung ab 1982, Patent 1988



Suzanne Vega

"Tom's Diner" (1987)

- Der erste Song, der als MP3 kodiert wurde
- A cappella (keine Instrumente)
- Klare, hohe Frequenzen
- Perfekter Stresstest für Kompression
- Brandenburg hörte "Tom's Diner" über 10.000 Mal



Wie funktioniert MP3?

Ein Zusammenspiel aus vielen Faktoren:

- **1. Frequenz-Analyse (FFT)**
Audio → Frequenzspektrum
- **2. Psychoakustisches Modell**
Welche Töne hört Mensch nicht?
- **3. Quantisierung**
Unwichtige Frequenzen reduzieren
- **4. Huffman-Coding**
Lossless-Kompression der Restdaten

Bitrate: Der Qualitäts-Knopf

Bitrate	Qualität	Kompression
128 kbps	Hörbar schlechter	~11x
192 kbps	Akzeptabel	~7x
256 kbps	Gut	~5,5x
320 kbps	"CD-Qualität"	~4,4x

Original CD: 1.411 kbps (unkomprimiert)

Der Patentkrieg

1990er: Fraunhofer + Thomson halten MP3-Patente

Lizenzgebühren:

- \$0,75 pro Decoder
- \$2,50 pro Encoder

Problem: Napster (1999) → unkontrollierte Verbreitung

2017: Patente laufen aus → MP3 ist frei

Napster (1999)

P2P-Filesharing für MP3s

- Shawn Fanning, 19 Jahre alt
- 80 Millionen Nutzende in 2 Jahren
- Musikindustrie verklagt (2001)
- Pandora's Box: Nicht mehr aufzuhalten

The screenshot shows the Napster website interface. At the top left is the Napster logo. To its right is a quote: "We should think of (Napster) as a new kind of radio--a promotional tool that can help artists who don't have the opportunity to get their music played on mainstream radio or on MTV..." -- Chuck D, with a link to < more >. Below the logo is a navigation menu with links: Home, Download, Speak Out, Discover, Press Room, Help, Service Status, Company, and Policies. A "Newsletter" section includes a "Your Email" input field and a "Subscribe" button. A "Call Congress!" button with a telephone icon is also present. The main content area features a "Speak Out!" section with a graphic of a person shouting and text explaining the recording industry's actions and a call to action to join the Napster Action Network. It includes links for "Join the Napster Action Network", "Legal Update Page", and "Contact Congress" (with a zip code input field and a "Find" button). Below this is a "Featured Music" section with a "New Artist Program" that includes the Napster logo and text about finding new music. At the bottom, there is a footer with navigation links: home | download | speak out | discover | press room | help, and a copyright notice: "Copyright 2001 Napster Inc. All Rights Reserved".

Napster & Musikindustrie

1999: Napster startet

2001: 80 Millionen Nutzende

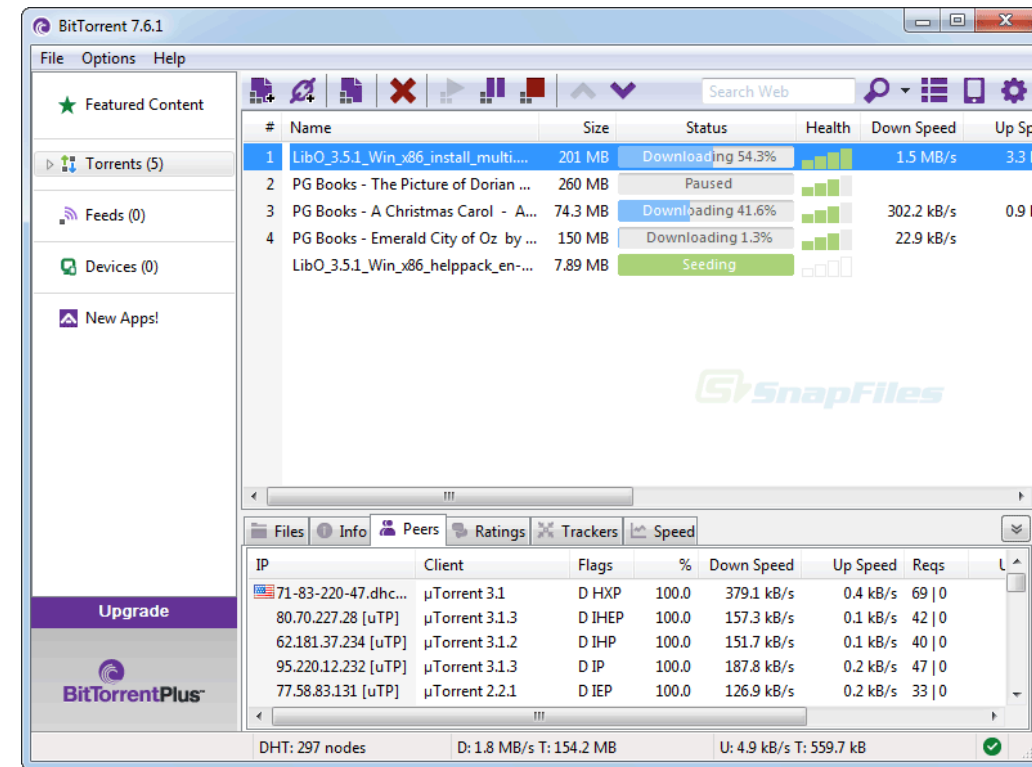
Musikindustrie:

- CDs kosten \$15–20
- MP3s gratis (illegal, aber yolo)
- Einzelne Songs statt Alben

2001: Napster wird verklagt und schließt

Aber: Pandora's Box offen

→ LimeWire, Kazaa, BitTorrent, später Spotify









Kulturelle Revolution

MP3 veränderte:

- ✓ Musik wurde portabel (Walkman → iPod)
- ✓ Alben wurden irrelevant (Playlists)
- ✓ Musikkonsum explodierte (kostenlos/billig)
- ✓ KünstlerInnen verloren Kontrolle

Aber auch:

-   KünstlerInnen verdienen weniger pro Stream
-   Audio-Qualität sank (Loudness War)
-   Physische Medien starben

Fragen & Diskussion

Kontakt: lb-czechowski@hdm-stuttgart.de

Folien: Online verfügbar unter <https://librete.ch/hdm/223015b>

Lizenz & Attribution

Diese Präsentation ist lizenziert unter **Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)**

- Erlaubt Teilen & Anpassen mit Namensnennung
- Adaptionen müssen unter gleicher Lizenz geteilt werden

Vollständige Lizenz: <https://creativecommons.org/licenses/by-sa/4.0/>

Selbstlernen: Audio-Spektrogramm

Aufgabe (30 Min):

- Live Spektrogramm untersuchen <https://borismus.github.io/spectrogram/>
- Mit Effekten experimentieren <https://audiomass.co/>
- Spektrogramme vergleichen Audacity (kostenloser Download nötig)
https://manual.audacityteam.org/man/spectrogram_view.html

Selbstlernen: HEX Files

1. Fünf Dateien ohne Dateiendung:

`hex1` · `hex2` · `hex3` · `hex4` · `hex5`

2. Lies erste 16 Byte aus und identifiziere Dateiformat (Magic Number)

3. *Optional: Datei umbenennen und korrekte Dateiendung anhängen (bspw. `.jpg`)*

Tools:

- Hex-Editor: hexed.it
- Magic Numbers: en.wikipedia.org/wiki/List_of_file_signatures

