

Einführung in die Medieninformatik

1. Semester

FurTactics*

03.04.2024

Contents

Allgemeines

1 Audio

1.1	Diskretisierung	2
1.2	PCM	2
1.3	Kompression	2
1.4	Klangsynthese	3
1.5	Hexadezimaldarstellung	3
1.6	Gesamtgröße der Datei	3

2 MCI

2.1	Interaktion	4
2.2	Anforderungsanalyse	4
	Card Sorting	4
	User-Centred Design (UCD)	4

3 Dart&Flutter

3.1	Dart	4
	Kontrollstruktur <code>switch</code> :	4
3.2	Apps	4
	Digitale Barrierefreiheit	5
	Gesten	5
3.3	Flutter	5
	Klassen	5
	Buttons	5

4 Evaluation

4.1	Heuristiken	5
	Farbkontraste	5
4.2	Evaluationsmethoden	5

TLX	6
-----	---

5 Bild

5.1	SVG	6
5.2	Licht	6
5.3	Farbmodelle	7
5.4	Digitale Bilder	7
5.5	Digitale Bildformate	7
5.6	Vektorgrafik	8

6 Wahrnehmung

6.1	Psychophysikalische Skalierung	8
6.2	Gestalt	9

7 Kodierung

7.1	Signale	9
7.2	Entropie	9
7.3	Umrechnung	9
7.4	Huffman und Fano	9

8 Text

8.1	Zeichensätze	9
8.2	Braille-Schrift	10
8.3	Schrift	10
	Eigenschaften	10
	Absätze	10
	Formate	10

9 XML

9.1	DTD	11
-----	-----	----

10 JSON

*Danila Dolgoplov danila.dolgoplov@mailbox.tu-dresden.de

Allgemeines

Informatik ist die Wissenschaft, die sich mit den theoretischen Grundlagen, den Mitteln und Methoden der Informationsverarbeitung unter Einsatz von Computern beschäftigt

Aufgabenstellungen

Berechenbarkeit (Lösen mathematischer Probleme), Unberechenbarkeit, Churchsche These (Gleichwertigkeit aller Maschinenkonzepte), Chomsky-Hierarchie (Zusammenhang Automaten \leftrightarrow Sprachgrammatik), Programmkomplexität, Programmverifikation

Medienkompetenz

Medialitätsbewusstsein + Medienwissen, Medienspezifisches Rezeptionsmuster, Medienbezogene Genussfähigkeit (Bedürfnis nach Identifikation und Unterhaltung), Medienbezogene Kritikfähigkeit.
Medien: Mittel zur Speicherung von Information (Physisch oder Format, Text, Video, Audio)

1 Audio

1.1 Diskretisierung

- **Diskretisierung:** Festes Raster von Messpunkten gleichen Abstands wird festgelegt
- Typischer Frequenzbereich 16 Hz - max. Hörfrequenz ca. 20kHz
- **Analoge Audio-Signale:** Periodische Signale (Wellen), Amplitude (maximaler Wert): Lautstärke [dB], Frequenz: Tonhöhe, Phase bezeichnet den Zeitraum der Wiederholung, Obertöne sind Vielfaches einer Grundfrequenz
- **Abtastrate:** die Dichte der Messwerte und wird als Samples pro Sekunde ausgedrückt
- **Quantisierung:** Rundung der Messwerte auf das Raster / Darstellung der Messwerte / Amplitude als Binärzahl
- **Auflösung** ist die Anzahl Bits pro Sample. Sample: Ein Wert / eine Abtastung

- **Abtastfehler:** Bei zu geringer Abtastrate kann sich ganz anderes Muster ergeben (konstanter Ton statt Schwingung)
- **Nyquist-Shannon-Abtasttheorem:**
 $f_{abtast} > 2 \cdot f_{max}$, damit alle Signale ohne Informationsverlust dargestellt werden können

1.2 PCM

- **PCM (Pulse Code Modulation):** Analoge Spannung in digitale Daten verlustfrei aber fehlerbehaftet umwandeln (analoges Signal über analoge Filterstufe über zeitliche Abtastung über Quantisierung über Codierung zum binären Digitalsignal)
- **Filter:** Tiefpass mit Grenzfrequenz f_s – höchste Signalfrequenz
- Analogsignal wird mit dem linearen 3-Bit Wandler (8 Stufen) wertdiskretisiert, 2er-Komplementbildung
- **Audio-CD:** 44100 Hz, 16 bit / Sample, 2 Kanäle (Stereo), d.h. Datenrate = $2 \cdot \frac{44100}{s} \cdot \frac{16 \text{Bit}}{8 \text{Byte}} = 10 \frac{\text{MByte}}{\text{min}}$. 12 cm Durchmesser, Umdrehungsgeschwindigkeit je nach Entfernung vom Mittelpunkt. Oberfläche besteht aus Pits und Lands (Löcher und Hügel). Hoch redundant, durch 16 bit pro Sample
- **Blu-ray disc:** Kürzere Wellenlänge als bei DVD und CD, bessere Fokussierung, 36 MB/s, 25 GB pro Layer, MPEG-2 kodiert, beidseitige Laserabtastung
- **RIFF (Resource Interchange File Format):** WAV (PCM) und AVI (AVID). Aufbau in Chunks (Chunk-ID, Chunk-Size, Chunk-Data)
- **WAV:** 3 Chunks. Header-Chunk: "RIFF" (4 Byte), Dateigröße 4 (8 Bytes weniger als im System angezeigt), "WAVE" 4, Format-Chunk: "fmt" 4, Größe des Formatblocks 4 (16), Formatblock: Format 2 (1 – PCM), Kanäle 2 (1 – Mono, 2 – Stereo), Samplefrequenz (Samples pro sec je Kanal) in Hz 4 (bspw. 44100), Byte pro Sekunde 4, Byte pro Sample (Framegröße) 2,

Bit pro Sample (Samplingtiefe) 2, Daten-Chunk: "data" 4, Größe des Datenblocks 4, Datenblock

- **PCM vs. Analog:** störresistente Übertragung möglich, verlustfreie Weiterbearbeitung, kostengünstige Schaltkreise verfügbar, aber Quantisierungsfehler (Rundung der Werte bei Einfügen in diskrete Tabelle) und Hintergrundrauschen bei der Aufzeichnung
- **Quantisierungsfehler** maximal mit Betrag von $\frac{1}{2}$ der Quantisierungsstufe, hörbar als Rauschen
- **Rauschen:** Einfluss der Codierung auf die Signalqualität, z.B. Quantisierungsfehler, ab 70dB unter Maximumpegel nicht hörbar
- **DPCM (Differential PCM):** Differenz zwischen benachbarten Abtastwerten werden quantisiert und codiert, Geringere Wortbreite bei gleicher Qualität (max. Kompressionsfaktor 1 : 2)
- **ADPCM (Adaptive Differential PCM):** periodisch neu berechnete Skalierung (leise: feine Einteilung, laut: grobe Einteilung), bessere Vorhersage durch Prädiktor. Versucht, den weiteren Signalverlauf vorherzusagen, und kodiert und überträgt dann nur die Differenz zwischen tatsächlichem und vorhergesagtem Signal. Steile Signale teilw. nicht vollständig reproduzierbar. Bei CD-Qualität 1 : 4-fache Kompression, bis zu 32-fache Kompression beim starken Qualitätsverlust
- Weitere Formate: **MP3** (verlustbehaftete Kompression), **MOD** (Atari Musik), **AU**
- **MIDI** (31250 Bit pro Sekunde), Byteweises Festlegen von Instrument, Tonhöhe, Dauer usw., Standardisiert, Erweiterbar

1.3 Kompression

- **Motivation:** Akustische Täuschung. Anforderungen:
 - Geringe Toleranz erforderlich (Gehör genau, Abspielrate darf nicht länger als 10ms abweichen, Signal darf nicht wiederholt / ausgesetzt werden für

2ms, Gehörmpfindlichkeit ist frequenzabhängig, Sprache: Inhalt wichtig [viel Semantik], bei Musik originalgetreue Wiedergabe wichtig)

- Hörempfindlichkeit ist frequenzabhängig - Isophone: Stufen gleicher Lautstärke

- **Wellenformcodierung** (ISDN, PCM, WAV): Einfache Technologie, Ziel: Möglichst exakte Nachbildung
- **Frequenzbereichcodierung** (MP3, WMA, OGG Vorbis): Komplexe psychoakustische Modelle, Frequenzbänder werden in unterschiedlicher Qualität übertragen, Ziel: Möglichst genaue Nachbildung beim Weglassen von möglichst viel Information
- **Parametrische Codierung** (GSM Sprachcodierung): Speicherung von Mustersequenzen (wie indexierte Bildformate), geeignet für Sprache (häufige (Teil)Widerholungen)
- **Spracherkennung** (noch unausgereift) Erkennung → Textübertragung → Synthese, dann auch auf Musik übertragbar
- **Redundanz**: Elemente ohne zusätzliche (neue) Information
- **Irrelevanz**: Nicht wahrnehmbare Elemente
- **Entropiecodierung**: Verlustfreie Kodierung, Eliminierung der Redundanz
- **Reduktion**: Eliminierung von irrelevanten Daten
- **Dekorrelation**: Umformung in andere Darstellung ohne Datenmenge zu ändern
- **Codec** (Encoder / Decoder): Kombination verschiedener der oberen Techniken zu einem komplexen Modul
- **Pipeline Encoder**: Signalaufbereitung (Abtasttheorem, Quantisierung), Signalzerlegung (Dekorrelation, Frequenzzerlegung o.ä.), Quantisierung (Perzeptionsmodelle, Reduktion), Entropiekodierung (Statische Modelle, z.B. ZIP)

- **Mithörschwelle**: Laute Töne maskieren leise Töne, Frequenzanteile unterhalb der Kurve weglassen (laute Töne maskieren bis zu 20ms vor dem Ton und bis 200ms nach dem Ton, abhängig von Lautstärke, Frequenzverhältnis, zeitlicher Lage, Struktur/Dauer des Maskierens)
- **Irrelevanzkodierung**: Herausfinden der Wahrnehmungsgrenzen, Anpassen der Quantisierung, Löschen der irrelevanten Teile. Dadurch mehr Quantisierungsrauschen, also Kompromiss aus Qualität und Bandbreite nötig
- Annäherung eines Rechtecksignals durch ungerade Vielfache einer Grundfrequenz
- **Frequenzspektrum**: Graph mit Pegel [dB] auf Y-Achse und Frequenz auf X-Achse
- **Interferenz**: Überlagerung von Wellen (Konstruktiv: Addierend, Destruktive: So, dass Welle sich ausgleicht)
- **Fourier-Transformation** (FFT) und **Diskrete Kosinus-Transformation** (DCT) erlauben die Umrechnung aus dem Signalraum (Frequenz über Zeit) in den Frequenzraum (Vorkommen von einzelnen Frequenzen)
- **Subband Codierung**: Filterbank zerlegt Signal in Frequenzverbänder, Gesamtenergie des Signals meist ungleichmäßig verteilt, viele Bänder mit wenig Energie, wenig Bänder mit viel Energie, pro Frequenzband Maskierung bezüglich Irrelevanz (Hörschwelle, Verdeckung) und Quantisierung (Bitrate variabel), Bitstrommultiplexer erzeugt digitalen Datenstrom (inverse Filterbank setzt Signal fast perfekt zusammen)
- **MP3**: Zerlegung in 32 Frequenzbänder, pro Teilband 18 Subbänder, Rauschen minimieren, Entropiekodierung, verdeckte Signale erkennen, optimale Codierung finden, 1152 Samples pro Frame, 30-230 kBit/s

1.4 Klangsynthese

- **Wave-Table ("Sampling") Verfahren**: Eine Periode wird für periodische Signale numerisch gespeichert und wiederholt abgespielt

- **Additive Synthese**: Addieren von mehreren (>100) Sinus und Cosinus-Signalen (harmonischen Teiltönen) unterschiedlicher Frequenz kann beliebige Wellenformen erzeugen (Fourier-Synthese)
- **Subtraktive Synthese**: Oszillator erzeugt ein obertonreiches Signal (z.B. Sägezahn), das durch Filter bearbeitet wird
- **FM Synthese ("Keyboard")**: Modulieren einer hörbaren Trägerfrequenzen durch eine hörbare Modulationsfrequenz erzeugt ebenfalls ein obertonreiches Signal, erlaubt günstige Synthesizer

1.5 Hexadezimaldarstellung

Z.B. die Abtastrate aus dem Datenblock in Hex-Editor ist 40 1F 00 00 → 00 00 1F 40 =
 $= 1 \cdot 16^3 + 15 \cdot 16^2 + 4 \cdot 16^1 + 0 \cdot 16^0 = 8000 \text{ Hz}$

1.6 Gesamtgröße der Datei

Abtastfrequenz [44100 Hz] · **Samplingtiefe** [$16 \frac{\text{Bit}}{\text{Sample}} = 2 \frac{\text{Byte}}{\text{Sample}}$] · Anzahl der **Kanäle** [2] + 44 Byte - Gesamtgröße in System **ODER** + 44 Byte - 8 Byte - Gesamtgröße in WAVE-Hex

2 MCI

MCI – Mensch-Computer Interaktion

- **Menschliche Informationsverarbeitung**: visual overload, motorische Fertigkeiten
- **Design für Diversität von Menschen**: Erstellen von Produkten, Gebäuden und Umgebungen die von vornherein zugänglich sind für so viel wie mögliche Benutzer, d.h. auch für ältere Menschen, Menschen ohne und mit Behinderungen
- **SehSchädigung**: Menschen mit SehSchädigung verwenden Sprachsynthese – Screenreader (Talkback für Android und VoiceOver für iOS), Spracheingabe und Gesten. Screenreader unterstützen auch Braillezeilen per Bluetooth
- **Farbenblind**: Nicht nur Farbe, sondern auch geometrische Figuren zu verwenden, um wichtige Informationen zu vermitteln

- **Hörbehinderung:** Untertitel und Spracherkennung, Sprachdolmetscher schreiben gesprochenes nach, Geräuscherkennung zur Visualisierung von akustischen Signalen
- **Körperbehinderung:** Saug-Blaseschalter, Zwinkern, Eyetracking, Kopfgeste
- **Ältere Menschen** (Sehebehinderung): Große Icons und Texten, höher Kontrast, Spracheingabe und Sprachausgabe

2.1 Interaktion

WIMP (Benutzungsschnittstellen) – Windows Icons Menus Pointers

Gebrauchstauglichkeit ist das Ausmaß, in dem ein Produkt, System oder ein Dienst durch bestimmte Benutzer in einem bestimmten Anwendungskontext genutzt werden kann, um bestimmte Ziele **effektiv**, **effizient** und **zufriedenstellend** zu erreichen

1. **Eingabe/Quelle:** Tastatur, Mikrophon, Kamera und Speichermedium (SSD, HDD, CD, DVD)
2. **Verarbeitungseinheit:** Konverter (Filter, Kompression), Mischer, Splitter
3. **Ausgabe/Senke:** Lautsprecher, Monitor, Drucker und Speichermedium (SSD, HDD, CD, DVD)

2.2 Anforderungsanalyse

Analyse der Aufgaben der Benutzer um kleinere Interaktionsschritte festzulegen

Card Sorting

Ermittelt wie Benutzer die Dinge benennen (Namen), gruppieren (Relationen) und die Gruppen benennen (Kategorien)

- **Geschlossenes Card Sorting:** mit vorgegebenen Gruppen (für Erweiterungen)
- **Offenes Card Sorting:** ohne vorgegebene Gruppe (für neues Design, aber führt zu sehr vielen Oberbegriffen)

User-Centred Design (UCD)

Kann man beliebig oft wiederholen: Notwendigkeit → Kontext festlegen und verstehen (Card Sorting) → Beschreibung des Benutzers → Erstellen des Design (Prototypen/Mockup) → Design evaluieren (Heuristik) → System erfüllt Benutzeranforderungen

Prototypen: Low Fidelity: Skizzen, Storyboard, Wireframe, Papier, Mitte: Mock-Up, Wizard-of-Oz, High Fidelity: Computereinsatz (Powerpoint), Implementierung der Interaktionstechniken

Wizard-of-Oz: eine Prototyping(Evaluations)-Technik, um technisch aufwändige Interaktionen vor deren Umsetzung zu evaluieren, bei der die Antworten des Systems durch eine Person (Wizard), statt durch die Software generiert werden

3 Dart&Flutter

3.1 Dart

Dart ist die objekt-orientierte Sprache und unterstützt Hot Reload. Dart läuft an den Virtuellen Maschinen (VM) und ist unabhängig von der Architektur des Prozessors (Intel/AMD). Dart ist single threaded mit event loop und microtasks und dynamisch, d.h. der Speicherbedarf wächst und schrumpft automatisch
Einfaches Beispiel für den Aufruf der Funktion **print:**

```
void main() {
  print('Hallo Welt'); /*print() ist eine
                       Anweisung (Statement)*/}
```

Deklaration der Variablen ist eine Bekanntgabe des Namens und des Typs (Basistyps und Klasse), z.B. `int i; String s; bool ok;`. Alle Variablen außer `String`, `bool` und `int` können eine Referenz auf `null` sein

Klasse ist ein abstraktes Modell bzw. einen Bauplan für eine Reihe von ähnlichen Objekten

Objekt ist ein Exemplar eines bestimmten Datentyps oder einer bestimmten Klasse (auch "Objekttyp" genannt)

```
class Square{
  double? length;
  double area(){return length!*length!;}}
```

```
void main(){
  Square quadrat = Square(); // Erstellung
                             eines neuen Elements
  quadrat.length=10;
  print("Area of rectangle is
        ${rectangle.area().}");}
```

Kontrollstruktur switch:

```
...
switch (opnd) {
  case "-":
    result = "Subtrahieren";
    break;
  default: /*verpflichtet*/
    result = "Unbekannter Operator";
    break;
}
```

Funktionen mit optionalen Parametern:

```
void func(String s, [int n=0])
```

Funktionen mit benannten Parametern:

```
void func(required String s, required int n)
und der Aufruf: func(s: "a", n: 1);
```

Listen: `var list = [1, 2, 3];` und die Länge ist `list.length`

Schleifen: Mit `if`-Anweisung in `while`-Schleife kann man nach einer Bedienung `break`; to stop looping und `continue`; to skip to the next loop iteration

Mit `do{};` `while()` kann man vor der Schleife etwas machen

3.2 Apps

Apps bestehen aus Widgets/Views, die Stateless oder Stateful sein können. Die wichtigsten Widgets sind Text, Textfield und Button

- **Ausgabetechnik:** Display, Lautsprecher, Vibrationsmotor
- **Eingabetechnik:** Multitouch-Eingabe, Sensoren (Mikrofon, Kamera, Lagesensor, Gyroskop usw.), Funkschnittstellen (Wi-Fi, Bluetooth, NFC usw.)
- **MEMS** (Micro-Electro-Mechanical-Systems) - Sensoren messen Bewegungen in eine Richtung

oder eine Kreisbewegung für z.B. Lageänderung, Schrittzähler oder Indoor Navigation

- **Annäherungssensor:** Display ausschalten bei Annäherung an Kopf während des Telefonierens
- **Magnetsensor** (Kompass): Messung des Magnetfelds der Erde

Digitale Barrierefreiheit

4 Kriterien zur Softwareentwicklung:

1. **Wahrnehmbarkeit** (z.B. statt Pixeln Bildbeschreibung, Formeln; Untertitel)
2. **Bedienbarkeit** (z.B. Gesteneingabe führt zu Link, Tastaturunterstützung)
3. **Verständlichkeit** (z.B. Gebärdensprache, Leichte Sprache)
4. **Robustheit** (trotz Javascript, Einsatz von ARIA)

Gesten

- Streichen mit 1-3 Fingern kurz, doppelt oder mit Wartezeit
- Schüttelgesten

3.3 Flutter

Flutter erweitert Dart um Widgets zur Entwicklung von interaktiven Benutzeroberflächen für Desktop, Browser und Mobiltelefone aus demselben Programm. Dafür nutzt er Klassen ("Blaupausen für Widgets" bestehen aus Datenstrukturen und Funktionen und beschreiben ein Objekt) und Objekte (konkrete Datenstrukturen und Funktionen)

Klassen

`MyApp()` ist ein Konstruktor, der ein Objekt der Klasse `MyApp` erstellt

```
import "package:flutter/material.dart";
```

```
void main() => runApp(MaterialApp(
  home: Center(
    child: RichText(
      text: TextSpan(
```

```
text: "Hello World",
),),),),));
```

Widgets: Align, Column, Container, Icon, Image, Scaffold, Text usw.

Buttons

- **Radiobutton:** Benutzer wählen genau eine von mehreren Alternativen
- **ToggleButton:** Benutzer schalten den Zustand eines Schalters um
- **Checkbox:** Benutzer aktivieren oder deaktivieren eines oder mehrerer Schalter

4 Evaluation

Bewertung eines interaktiven Systems kann durch Experten oder Benutzer realisiert werden. Ziel ist die Feststellung der Qualität der Benutzungsoberfläche nach der Gebrauchstauglichkeit

4.1 Heuristiken

Verfahren, um Probleme zu lösen bei begrenztem Wissen und wenig Zeit

Heuristik zur Barrierefreiheit: Wahrnehmbarkeit, Bedienbarkeit, Verstehbarkeit und Robustheit

Farbkontraste

Wahrnehmung von Farbkontrasten wird beeinflusst durch Farbton, Helligkeit und Sättigung. Um den Kontrast zu definieren, verwendet man ein Farbmodell und vergleicht benachbarten Farbwerte

Standard RGB (Red, Green, Blue) (sRGB) definiert die Farben derart, dass eine der Farbkomponenten maximal sein kann und die anderen Null sind, um die entsprechende Farbwirkung zu erzielen. Also Farbwerte $0 \dots 255$ (8 bit) bzw. $00_{16} \dots FF_{16}$

Der Farbkontrast von zwei Farben laut der WCAG (Web Content Accessibility Guidelines) ist dann ausreichend, wenn ihr Helligkeitswert L_1 und L_2 das Verhältnis von $t = 3$ oder besser aufweist wobei

$$t = \frac{L_1 + 0,05}{L_2 + 0,05}$$

- $t \geq 4,5$ für normale und kleine Schrift (14 pt) für Level AA nach WCAG 2.0
- $t \geq 3$ für große oder fette Schrift (18pt bzw. 14 pt fett) für Level AA nach WCAG 2.0
- $t \geq 7$ für normale und kleine Schrift (14 pt) für Level AAA nach WCAG 2.0

Berechnung der Luminanz L im sRGB Farbraum:

$$L = 0.2126 \cdot \mathbf{R} + 0.7152 \cdot \mathbf{G} + 0.0722 \cdot \mathbf{B}$$

$$R_{sRGB} = \frac{R_{8bit}}{255}, G_{sRGB} = \frac{G_{8bit}}{255}, B_{sRGB} = \frac{B_{8bit}}{255}$$

$$\mathbf{R} := \begin{cases} \text{if } R_{sRGB} \leq 0.03928 & R = \frac{R_{sRGB}}{12.92} \\ \text{else} & R = \left(\frac{R_{sRGB} + 0.055}{1.055} \right)^{2.4} \end{cases}$$

$$\mathbf{G} := \begin{cases} \text{if } G_{sRGB} \leq 0.03928 & G = \frac{G_{sRGB}}{12.92} \\ \text{else} & G = \left(\frac{G_{sRGB} + 0.055}{1.055} \right)^{2.4} \end{cases}$$

$$\mathbf{B} := \begin{cases} \text{if } B_{sRGB} \leq 0.03928 & B = \frac{B_{sRGB}}{12.92} \\ \text{else} & B = \left(\frac{B_{sRGB} + 0.055}{1.055} \right)^{2.4} \end{cases}$$

4.2 Evaluationsmethoden

Qualitäts- bzw. Gütekriterien für die Evaluation sind Objektivität (sind die Ergebnisse unabhängig vom Versuchsleiter?), Validität (gleiche Ergebnisse für denselben Gegenstand?), Reliabilität (mißt der Fragebogen was er messen soll?)

- **Messgrößen:** Graphen, Zählungen (quantitative und thematische Analyse), Muster erkennen (qualitative Analyse)
- **Diagrammarten:** berichten über Daten und erfinden keine zusätzlichen Daten
- **Graphen aus mehreren Werten:** Gestapelte Balkendiagramme für bspw. Höchst-, Tiefst- und Mittelwerte
- **Deskriptive Statistik: Median:** max. 50% Daten $< z$, min. 50% Daten $\leq z$ max. 50% Daten $> z$, min. 50% Daten $\geq z$
- **Fragebogen:** Vorgehensweise: Hypothese formulieren, Klären die Variablen und Merkmale, Klären welche Messwerte erhoben werden können

TLX

TLX ist ein NASA Task Load Index zur Bestimmung der kognitiven Workload. Der kognitive Workload ist die mentale Belastung, die mit einer Aufgabe verbunden ist.

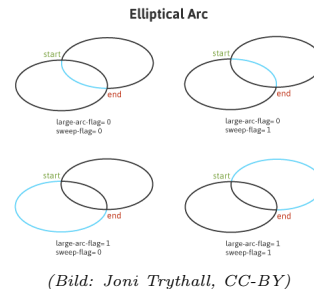
Der TLX Fragenkatalog ermittelt die subjektive mentale Belastung (Workload) bei der Bearbeitung einer Aufgabe oder im Umgang mit einem System. Teilnehmer beantworten 6 Fragen zur Effektivität: *Mental Demand* (MD): How mentally demanding was the task? *Physical Demand* (PD): How physically demanding was the task? *Temporal Demand* (TD): How hurried or rushed was the pace of the task? *Performance* (OP): How successful were you in accomplishing what you were asked to do? *Effort* (EF): How hard did you have to work to accomplish your level of performance? *Frustration* (FR): How insecure, discouraged, irritated, stressed, and annoyed were you?

5 Bild

5.1 SVG

- EncapsulatedPostScript, Bildinformationen in speziellem Code, programmierbar, keine Alphakanäle, nur PostScript-Geräte
- **SVG** (Scalable Vektor Graphics) in XML (Extensible Markup Language) für Bilder, Text, Animation, Formen (bspw. mit Inkscape malbar)
- **Primitive:** circle-Element mit Attributen cx, cy und r (CenterX und CenterY), ellipse mit cx, cy, rx und ry, rect mit x, y, width, height, line mit x1, y1, x2, y2 ... , polygon / polyline mit points = "x1 y1 x2 y2 ..."
- **Pfade:** `<path d=""></path>`, Möglich anzugeben: **M** x y (Move To), **L** x y (Linie zum Ziel malen), **H** x (Horizontale Linie), **V** y (Vertikale Linie), **Z** (Pfad schließen), **C** x1 y1 x2 y2 x y (wo x1 y1 x2 y2 – Kontrollpunkte und x y – Endpunkt) (Kubische Bézier-Kurve), **S** x2 y2 x y (wo x2 y2 – Kontrollpunkt und x y – Endpunkt) (Kubische Bézier-Kurve fortsetzen, vorherigen Kontrollpunkt spiegeln), **Q** x1 y1 x y

(Quadratische Bézier-Kurve), **T** x y (Quadratische Bézier-Kurve fortsetzen, vorherigen Kontrollpunkt spiegeln), **A** rx ry x-axis-rotation large-arc-flag sweep-flag x y (wo rx ry – Radien, large-arc-flag: der kurze (0) oder längere (1) Weg, sweep-flag: der obere (1) oder untere (0) Bogen und x y – Endpunkt) (Ellipsenbogen) - Groß geschriebene Buchstabe steht für absolute, kleine für relative Koordinaten. Pfade beginnen immer mit moveto (M) und können mit Z enden



- **Text:** text-Element mit x, y, font-size
- **Image:** image-Element mit x, y, width, height, xlink:href
- **Einfärben** von Formen mit fill-Attribut und einer Farbe in Hex (z.B. #FF0000) oder Name.
- **Rand** malen mit stroke-Attribut (enthält Farbcode) und stroke-width-Attribut (Randbreite in Pixeln)
- **transform**-Attribut gibt Rotation, Transformation, Skalierung oder Stauchung an: z. B. "translate(20, 150) scale(0.5) rotate(90,20,150) skewX(25) skewY(25)" - Bei Rotate dienen der zweite und dritte Parameter dabei als Rotationszentrum
- Clipping: Beschränken eines zu zeichnenden Elements auf den Bereich, den ein anderes Element definiert
- Filter in SVG: "Fotorealismus" durch Unschärfe, gezielte Einbringung von Bildstörungen o. ä.
- Blurring: Unschärfefilter

- Events in SVG mit JavaScript definieren (svg onload-Event, onclick-Event etc.)

• Animation

```
<svg>
  <ellipse id="elli" cx="100"
    cy="100" rx="48" ry="90"
    style="fill:green;">
  <animate attributeType="XML"
    attributeName="rx"
    begin="0s" dur="6s"
    from="48"
    to="198" fill="freeze" />
</ellipse>
</svg>
```

- In `<defs>...</defs>` können SVG-Elemente definiert werden, die nicht gezeichnet werden, sondern mit `<use xlink:href="#id">` später wieder benutzt werden

5.2 Licht

- Licht ist eine in Photonen gequantelte, transversale, elektromagnetische Welle
- **Lichtgeschwindigkeit** im Vakuum $c = 299792458 \frac{km}{s}$ ist vom Medium, in dem sich das Licht bewegt, abhängig (meist Luft), $c = \lambda \cdot f$ (λ : Mediumskonstante, f : Frequenz des Lichts)
- Frequenz ist Maß für **Energie** $E = hf = \frac{hc}{\lambda}$ (h ist das Plancksche Wirkungsquantum mit ca. $h = 6.626 \cdot 10^{-34} J \cdot s$), d.h. je höher die Frequenz, desto energiereicher ist das Licht
- **Frequenz** des sichtbaren Lichts beträgt ca. 400 nm - 780 nm, davor liegen die Frequenzbereiche von UV-Licht, Röntgen- und Gammastrahlung, darüber die von infrarotem Licht, UKW-Radio etc
- Je höher die Frequenz, desto geringer die Wellenlänge (antiproportional)
- **Absorption:** Bspw. beim Durchlaufen einer farbigen Folie wird ein Teil des Lichtspektrums absorbiert

- **Reflexion:** Ein Teil des Lichts wird beim Auftreffen auf eine Oberfläche reflektiert
- **Additive Farbmischung:** Überlagerung von Licht mehrerer Emissionsquellen, Spektren werden addiert
- **Subtraktive Farbmischung:** Einfallendes Licht fällt auf Filter (Absorption / Reflexion) ("Multiplikation mit Filterfunktion")
- Gedruckte Farben erscheinen je nach Beleuchtung unterschiedlich (z.B. Geld wird rötlich bei roter Beleuchtung)
- **Farbtemperatur in Kelvin:** "Wärme" des Lichtes, je höher desto weißer – Glühlampe ca. 2800K, Tageslicht: 6500K
- Mensch "sieht" Lichtintensität, visuelle Wirkung, Kontrast, Formen, Bewegung
- Sehen bzw. Licht beeinflusst Wachsamkeit, Wohlbefinden, innere Uhr usw.
- **Retina** im Auge mit Photorezeptoren besetzt, **Fovea** im Zentrum des Sehfeldes, "**Blinder Fleck**" des Auges ca. 1,75mm im Durchmesser (Gehirn "vervollständigt" das Bild)
- Farbwahrnehmung in der Retina erfolgt durch Zapfen und Stäbchen (Zapfen: Farbwahrnehmung, Stäbchen: Schwarz/Weiß)

5.3 Farbmodelle

- Neben Helligkeit nimmt man als Hauptmerkmale Farbton und Sättigung wahr
- Hirn teilt wahrgenommene Farben in weiß, grau, schwarz und verschiedene Farbtöne
- **Farbwahrnehmung:** Helligkeit (Luminance), Farbton (Hue) und Sättigung (Saturation) hauptsächlich
- **RGB-Farbraum:** 256 Farbstufen pro Rot, Grün und Blau je 1 Byte, also $256^3 \approx 16,7$ Millionen Farben (Truecolor)
- CIE 1932: xyz-Kanäle, y: Wahrgenommene Helligkeit, x: rot/grün, z: blau/gelb-Unterschied

- 128 Farbtöne (Hues), 130 Farbsättigungen (Saturation Levels), 16 (blau) - 26 (gelbe) Helligkeitswerte, also ca. 7 Millionen wahrnehmbare Farbnuancen
- **Munsell Farbsystem** (Farbton hue, Helligkeit value, Sättigung chroma)
- **HLS Farbsystem** (Hue, Luminance, Saturation)
- **CMY Farbraum** (Cyan, Magenta, Yellow): $(r, g, b) = (1, 1, 1) - (c, m, y)$
- **CMYK:** CMY mit Schwarz, damit Schwarz nicht aus Farbe gemischt werden muss (für Drucker toll)
- **YUV:** Y: Helligkeit (s/w), U, V: Farbbalance
- YC_bC_r : Y: Helligkeit, b: Blau, r: Rot
- Farben in Text: Hervorhebung, sparsam (auffällige) Farben einsetzen

5.4 Digitale Bilder

Digitale Bilder bestehen aus Pixeln ($w \frac{\text{Pixel}}{\text{Zeile}} \cdot h$ Zeilen), die pro Pixel meist 1-32 Bit Farbinfos beinhalten (Farbtiefe des Bildes) (ab 24 bit: **TrueColor**)

- **Indizierte Farbformate:** Zu Beginn des Bildes Farbtabelle, danach wird jede Farbe mit einem Index auf die Farbtabelle angegeben (sinnvoll bei wenigen verwendeten Farben)
- **Chroma Subsampling** bei YUV und YC_bC_r (4 : 2 : 2): Halbe Auflösung für g/b (U) und r/gr (V)-Kanäle
- **Alphakanal:** Transparenzangabe
- dpi/ppi: dots per inch/pixels per inch (Pixel pro Zoll, 1 *inch* $\approx 2,54$ cm)
- Monitor ca. 72 dpi, Laserdrucker ca. 1200 dpi

5.5 Digitale Bildformate

- **Histogramm:** Gibt Verteilung der Graustufen von 0 bis 1 des Schwarzwertes an

- **Bildfilter:** Beispielsweise Weichzeichner (lineare Mittelwertbildung aus Nachbarn um einen Pixel herum)
- **GIF** (Graphics Interchange Format): Animierbares farbindiziertes Bildformat mit 256 Farbstufen (Farbtiefe bis zu 8 Bit), kein Alphakanal, verlustbehaftete Kompression
- **PNG** (Portable Network Graphics): verlustfreie Kompression als GIF, Alphakanäle, freies Format, keine Animation, Farbtiefe bis zu 48 Bit, nicht für Fotos geeignet (verlustfreie Kompression zu schwach), Interlacing (Zunächst werden nur einige Zeilen des Bildes übertragen und dadurch eine Vorschau möglich, bevor das ganze Bild übertragen ist)
- **JPEG** (Joints Photographic Experts Group): Verlustbehaftete Kompression u.a. mit Huffmann-Kodierung, Bildung von 8x8-Blöcken – führt zu Blockbildung, JPEG-2000 als Nachfolger mit geringerer Artefaktbildung – Farbraumumrechnung in YUV, Tiefpassfilterung & Unterabtastung der Farbdifferenzsignale U, V (verlustbehaftet 4 : 2 : 2, s.o.), Quantisierung, Umsortierung, Huffmann-Kodierung. Gut geeignet für Fotos, nicht gut für alles andere (Diagramme etc)
- **JPEG2000:** basiert auf XML, ermöglicht Rechtemanagment, verschieden starke Komprimierung in verschiedenen Bildteilen, Interlacing-Support
- **TIF** (Tagged Image Format): mehrere Farbräume wie RGB, LAB, verlustfreie Komprimierung mit LZW-Algorithmus (Lempel, Ziv, Welch), können von den vielen Browsern nicht dargestellt/verarbeitet werden
- **PSD** (Adobe Photoshop): Kanäle, Bearbeitungsschritte, Text wird nicht gerastert, kann von vielen Programmen nicht gelesen werden

5.6 Vektorgrafik

- Beschreibung des Bildes mittels Formen wie Kreisen (Radius), Rechtecken (Höhe und Breite), Liniensegmenten (Endpunkte) usw. und nicht durch Pixel (Bsp.: SVG)

- **Transformationen** (Rotation, Translation, Skalierung, Scherung) können mittels der Multiplikation des alten Vektors mit einer Matrix und der Addition eines neuen Vektors ausgedrückt werden

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}$$

$$\text{Translation: } \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

$$\text{Rotation: } \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\text{Scherung: } \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\text{Skalierung: } \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

- **Bezierkurven** dritten Grades / kubische Kurven: Kurven nach dem Format: $\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = (1-t)^3 \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + 3t(1-t)^2 \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + 3t^2(1-t) \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} + t^3 \begin{pmatrix} x_3 \\ y_3 \end{pmatrix}$ mit $t \in [0, 1]$. Dabei sind $\begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$ und $\begin{pmatrix} x_3 \\ y_3 \end{pmatrix}$ die Start- bzw. Endpunkte der Kurve und $\begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$ sowie $\begin{pmatrix} x_2 \\ y_2 \end{pmatrix}$ definieren die Tangenten an diesen Punkten.

- **de Casteljau-Schema:** $t = \frac{1}{4}$
- **Gruppierung** von einzelnen Formen, damit Transformation auf alle Elemente in der Gruppe angewendet werden können
- **Tiefe:** Definierung einer Tiefenkoordinate, damit die Überlagerungsreihenfolge bei verdeckenden Formen definiert werden kann
- **TrueType-Fonts:** Die Kontur von Zeichen werden durch Bezierkurven beschrieben, wird dann gefüllt

6 Wahrnehmung

- **Meta Font (L^AT_EX):** Glyphenkontur wird durch interpolierende Kurve beschrieben

- "Wahrnehmung ist der Prozess, durch den Lebewesen Kenntnisse über ihre Umgebung und über sich selbst in Beziehung zur Umgebung gewinnen. Sie ist der Anfang allen Wissens."

- **Motivation:** Notwendige Daten an menschliche Wahrnehmung anpassen, um Platz zu sparen (z.B. nicht wahrnehmbare Töne bei Audio). Ergonomisch physische Dinge gestalten, Gestaltung von Benutzerschnittstellen, Layout etc. auf Menschen optimieren

- \leftrightarrow Reize \leftrightarrow (Psychophysik) \leftrightarrow Wahrnehmung \leftrightarrow (Psycho-, Neuro- und Biophysikologie) \leftrightarrow Neuronale Prozesse (Physiologie) \leftrightarrow

- **Funktionale Physiologie:** Erkenntnisse durch Patienten mit verletzten Gehirnen, elektrische Störungen von außen, durch teilweise Einschläferung, mit Elektro-Enzephalographie, Magneto-Enzephalographie

6.1 Psychophysikalische Skalierung

- **Wahrnehmung ist relativ:** Größe / Helligkeit
- **Auflösung des Auges:** δ minimaler Sehwinkel, bei dem 2 Linien im Abstand s getrennt erkannt werden

$$s = 2a \cdot \tan \frac{\delta}{2}$$

mit Betrachtungsabstand a

- Wahrnehmung der Länge erfolgt logarithmisch (bei längeren Strukturen müssen Unterschiede größer sein)

- Wahrgenommene Skalen für Größe von Kreisen ist abhängig nach vom Radius

- **Reizschwelle R_0 :** kleinster wahrnehmbarer Reiz (z.B. Helligkeit des dunkelsten, noch erkennbaren Sterns am Nachthimmel), quantitatives Maß (z.B. Pa beim Hören)

- **Unterscheidungsschwelle ΔR :** kleinster Unterschied von einem Grundreiz (z.B. Helligkeitsdifferenz für einen Punkt, den man gerade noch auf einem grauen Hintergrund wahrnehmen kann), qualitatives Maß

- **Messverfahren** erfordern Wahrnehmung von Reizen: Methode konstanter Reize (Zufällige Wiedergabe von verschiedenen starken Reizen, der zu 50% erkannte Reiz ist die Reizschwelle – sehr genau, langsam), Herstellungsmethode (VP stellt den Reiz selbst ein – ungenau, schnell), Grenzermethode (Reiz erhöhen bis ihn VP erkennt – genau, langsam)

- Empfindungsstärke Ψ , Intensität des Grundreizes R_0 , Unterscheidungsschwelle ΔR , k von Sinnesmodalität abhängige Konstante und c – Proportionalitätskonstante für die Skalierung

- **Webersche und Weber-Fechnersche Gesetze:**

$$\frac{\Delta R}{R} = k \text{ und } \Psi(R) = c \cdot \lg \frac{R}{R_0}$$

- Quadriert man die Stärke des Grundreizes, dann verdoppelt sich die Empfindungsstärke

- **Stevensches Potenzgesetz:**

$$\Psi(R) = c \cdot (R - R_0)^n$$

Entscheidend ist der Parameter n . Für Helligkeit gilt $n = 0.3$, für ein Gewicht $n = 1.5$, für die Länge einer Linie $n = 1.0$

- Empfindungsstärken wie Lautstärke: Verwendung einer logarithmischen Einheit wie Dezibel (Konstante k wird weggelassen)

- Eine Erhöhung um 10dB bewirkt eine Verdopplung der Lautheit

- **Bel:** Logarithmus des Quotienten zweier Leistungen

$$\text{Pegel} = \lg \frac{R}{R_0} \text{ [Bel]}$$

- **Dezibel:** Zehntel eines Bels

$$\text{Pegel} = 10 \cdot \lg \frac{R}{R_0} = 10 \cdot \lg \left(\frac{A^2}{A_0^2} \right) = 20 \cdot \lg \left(\frac{A}{A_0} \right)$$

[dB] und $A_0 = 2 \cdot 10^{-5}$ Pa heißt der Luftschall

6.2 Gestalt

- **Wahrnehmung:** Extraktion elementarer Merkmale (Kanten, Farbe) → Präattentive Verarbeitung (Erkennen elementarer Bestandteile) → Attentive Verarbeitung (Konstruktion des Gesamten)
- Wörter eines Textes werden nicht Buchstabe für Buchstabe wahrgenommen (Auge liest 6-8 Buchstaben und springt dann weiter (ca. 20 ms)), Hören funktioniert ähnlich (Speicherdauer ca. 1,5 s)
- **Gesetz der guten Gestalt/Prägnanz:** Übergeordnetes Gesetz zur Wahrnehmung von Gegenständen, Zusammenführung von prägnanten Einzeleindrücken zu Gestalt
- **Gesetz von Figur und Grund:** Figur wird als vor dem Hintergrund stehend wahrgenommen, Figur und Grund kann nicht gleichzeitig wahrgenommen werden
- **Gesetz der Gleichheit:** Ähnliche Dinge erscheinen zu zusammengehörigen Gruppen geordnet
- **Gesetz der Nähe:** Dinge, die sich nebeneinander befinden, erscheinen als zusammengehörig
- **Gesetz der Geschlossenheit:** Von Linien umschlossene Gegenstände werden als zusammengehörig empfunden
- **Gesetz des weiterführenden oder gleichen Verlaufs:** Bruchstücke reichen zum Bilden von Gestalten aus
- **Gesetz der Symmetrie:** Symmetrische Gebilde werden als Figur wahrgenommen
- **Gesetz des gleichen Schicksals:** Objekte, die sich mit gleicher Geschwindigkeit bewegen, werden als zusammengehörig empfunden
- **Gesetz der Erfahrung:** Wiedererkennen von bestimmten Formen
- Gestaltgesetze betreffen die präattentive Wahrnehmungsphase und finden vor allem z.B. bei der Interfacegestaltung Anwendung (Gestaltung von Icons, Bedienoberflächen)

- Bei rivalisierenden präattentiven Prozessen kommt es zu Wahrnehmungstäuschungen, Linien werden gebogen oder dynamische Effekte werden erzeugt

7 Kodierung

7.1 Signale

Ein **analoges Signal** ist die Änderung einer physikalischen Größe entsprechend einem Messwert der zu übertragenen Information

Ein **digitales Signal** orientiert sich an einem festen Raster und gibt Werte aus einem "endlichen Vorrat" möglicher Werte an (diskreter Wert)

Analoge Signale sind störanfällig gegenüber Rauschen (Schallplatte, Rundfunk, Foto). Digitale Signale sind unempfindlich gegenüber gegen Störsignale, solange das Signal nicht vollständig verfälscht wird oder ausfällt

Bitgruppen: 1 bit: Bit, 3 bit: Triade (Oktalsystem), 8 bit: Byte, 16 bit: Word, 32 bit: Doubleword, 64 bit: Quadword

7.2 Entropie

Entropie H ist ein Maß für den durchschnittlichen Informationsgehalt einer Nachrichtenquelle. Es gilt:

$$H = - \sum_i p_i \cdot \log_2(p_i)$$

Wobei p_i Auftrittswahrscheinlichkeit des Zeichens $g_i \in A$ ist und A das Alphabet. Maßeinheit ist 1 Bit.

7.3 Umrechnung

Dezimal zu Hexadezimal: Zahl durch 16 teilen, Rest als Ziffer nehmen. Dann das Ergebnis der Zahl durch 16 für den nächsten Schritt verwenden (also erneut durch 16 teilen). Zahl am Ende rückwärts nehmen

Hexadezimal zu Dezimal: Zahl von rechts nach links nehmen, erste Ziffer mit 16^0 multiplizieren, zweite Ziffer mit 16^1 usw. 2^{10} – Kibibyte, 2^{20} – Mebibyte, 2^{30} – Gibibyte usw.

7.4 Huffman und Fano

- **Alphabet:** Definiert lineare Ordnung auf Zeichen. Ein Binärcode ordnet jedem Element eines

Alphabets einen Binärcode zu

- **Fano-Bedingung/Präfixfreiheit:** Kein Zeichen ist Beginn eines anderen Zeichens
- **Mittlere Länge** eines Binärzeichens i mit der Länge N_i wird berechnet als:

$$L = \sum_i p_i \cdot N_i$$

- **Shannon's Theorem:** Es gilt immer $H \leq L$, Differenz kann sehr klein sein, die Differenz

$$R = L - H$$

wird **Redundanz** bzw. Nachrichtenanteil, der nicht unmittelbar zur Information beiträgt genannt

- **Fano-Kodierung:** Zunächst werden die Zeichen so auf 0 und 1 verteilt, dass die Summe der jeweiligen Wahrscheinlichkeiten möglichst gleich groß ist. Dies wird dann so lange wiederholt, bis jedes Zeichen einen eigenen Code hat
- **Huffman-Kodierung:** Es wird ein binärer Baum angelegt, dessen Blätter zunächst das Alphabet sind. Die beiden Zeichen mit der geringsten Wahrscheinlichkeit werden genommen und mit 0 und 1 beschriftet

8 Text

8.1 Zeichensätze

Zeichensatz definiert Zuordnung von Zahlen zu Zeichen

1. **ASCII:** 7-bit-Zeichensatz (128 Zuordnungen), 0-31 Steuerzeichen. Neuere Zeichensätze sind meist abwärtskompatibel (Erste 128 Zuordnungen entsprechen ASCII)
2. **ISO-8859-1...16:** 8-bit-Zeichensätze (256 Zuordnungen), 0-127 ist ASCII, Sonderzeichen wie deutsche Umlaute in allen Latin-Varianten an selber Stelle (-1: Latin1 westeuropäisch, -2: Latin2 osteuropäisch, -5: Kyrillisch, -6: Arabisch, -7: Griechisch, [...], -15: Westeuropäisch mit Eurozeichen)

3. **Unicode:** Internationaler Zeichensatz. Verschiedene Standards: UCS-2 mit 2 hoch 16 Bit Zeichen, UCS-4 mit doppelt so vielen Ebenen. Unicode 5 kennt 99 089 Zeichen. Formate:

- **BigEndian:** Höchstes Byte steht ganz rechts (TCP/IP, JPEG, PNG),
- **LittleEndian:** Höchstes Byte steht ganz links (Intel/x86, USB, PCI)
- **UTF-32:** Immer 4 Bytes für ein Zeichen
- **UTF-16:** 2 Bytes pro Zeichen mit Möglichkeit für mehr, wenn in ersten 2 Bytes spezielle Surrogate-Kodierung benutzt wird
- **UTF-8:** Wichtigstes Unicode-Format, weit verbreitet z.B. für XML. Besteht aus einem Byte pro Zeichen, das ASCII ist, wenn aber das oberste Bit gesetzt ist, dann ab 128 mit 2 Bytes, ab 2048 – 3 Bytes und ab 65536 – 4 Bytes

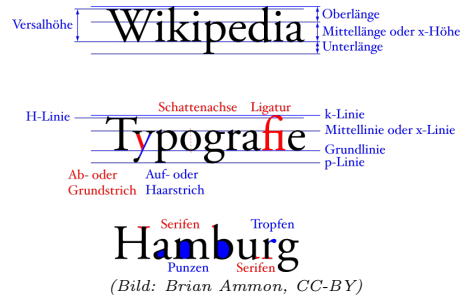
Aufbauschema: Höchstes Bit ist nicht gesetzt, wenn es nur ein ASCII-Zeichen ist und somit nur aus einem Byte besteht. Ist das höchste Bit gesetzt, gibt die Anzahl der dann folgenden Bits die 1 sind an, aus wie vielen Bytes das Zeichen besteht. Ist das erste Byte z.B. 11100000, folgen noch zwei weitere Bytes für das gleiche Zeichen.

8.2 Braille-Schrift

Von Louis Braille 1820 für Blinde entwickelt, für viele natürliche Sprachen verfügbar. Gibt Zeichen als fühlbare Punkte auf dem Papier wieder (6 - 8 Punkte kodieren ein Zeichen). Spezielle Schreibweisen für Formeln, Musik o.ä.

8.3 Schrift

Eigenschaften



- Höhe wird in Punkt angegeben. Ein Punkt ist meist 1/72 Zoll (TeX, DTP/Postscript), 1 Pica sind 12 Punkte.
- **Schriftstärke** (mager, normal, fett), **Schriftbreite** (schmal, normal, breit), **Schriftlage** (normal, kursiv)
- **Vektorschriften:** Definition durch Vektoren, **Rasterschriften:** Definition durch Pixelgrafik für je ein Zeichen
- **Serifen:** Helfen bei Lesbarkeit, sollte nur bei Fließtext benutzt werden
- **Kerning:** Anpassung von Zeichenabständen bei parallelen Diagonalen ("BRAVO": Zusammenrücken von A und V), Definition pro Buchstabenpaar. Kann doof sein bei geringen Abständen

Absätze

Absatzformat gibt Abstände und Einrückungen an. Möglich sind z.B. Blocksatz (Automatische Anpassung des Wortabstandes, sodass stets ganze Zeile gefüllt wird), Flattersatz/Mittelachsensatz (optimale Wortabstände). Seitenlayout: Kopf, Rumpf, Fuß

Formate

Eigenschaften: Editierbar? WYSIWYG (What You See Is What You Get)? Lesbar? Medien einbindbar? Funktionsumfang? Programmierbar? Barrierefreiheit?

- **Plain**

- **Rich Text .rtf** (Microsoft), eigene Definition für Medien, nicht programmierbar, alt, MS Word ist WYSIWYG-Editor

```
{\rtf Hallo \par {\i Dies} war kursiv.
\par Noch ne Zeile.}
```

- **LaTeX .tex** (frei), voll programmierbar, nicht WYSIWYG (wirft PDF o.ä. aus), Medien einbindbar, gute Formelsetzungsbibliotheken, vielseitig und toll, aber kompliziert
- **PDF** (Adobe), baut auf Postscript auf, Layout-Treu, Rechtemanagment, Einbetten von Schriften problematisch, kaum editierbar, wird aus anderen Formaten generiert. PDF/A ist archivierbar, PDF/UA ist barrierefrei
- **HTML:** Trennung von Struktur und Style mit CSS, Scripting (Document Object Model: Abbildung des XML-Baums auf JavaScript), programmierbar, kein WYSIWYG, Plugins

9 XML

- Medien in Dokumenten definieren sich durch **Inhalt** (Semantik), **Struktur** (Syntax) und **Präsentation** (lexikalische Information)
- Für Arbeiten mit HTML bedient man sich der Sprache, Werkzeugen und Arbeitsmethoden (Layouten o.ä.). HTML basiert auf XML, das als Metasprache (eine Sprache, um Sprachen festzulegen) Text beschreibt und strukturiert (Baumstruktur)
- Dokumente bestehen aus **Tags**
<person>Der Bart</person> oder für leere Elemente
, Elementname (person) legt das Tag als Ganzes fest. Metazeichen von XML sind entsprechend zunächst <, / und >
- **XML-Syntax:** Elemente können Elemente (Schachtelung) oder Text enthalten, Groß- und Kleinschreibung spielt eine Rolle und das öffnende Tag muss genauso lauten wie das Schließende

- In XML-Dokumenten immer **XML-Deklaration** am Dokumentenbeginn: `<?xml version="1.0">`. Folgen geht dann ein Stylesheet und eine DTD sowie die Daten (Elemente, Attribute und Text) sein
- In der XML-Deklaration kann als **Attribut** z.B. `encoding="iso8859-1"` (oder `"US-ASCII"`, UTF-8 ist voreingestellt, hier wird US-ASCII verlangt) und `standalone="no"` (gibt an, dass die DTD im Dokument selbst drin ist) verwendet werden
- **Attribut:** ist ein Paar `Attributname="Wert"`
- **Wohlgeformtes XML** ist syntaktisch korrekt (s.o., mit korrekten Schachtelungen, richtig verwendeten Metazeichen etc), **Valides XML** erfüllt die angegebene DTD und wohlgeformt ist
- **Wichtige Begriffe:** Elemente, Knoten (Tags), Vorgänger, Nachfolger, Nachbarknoten, Wurzel, Attribute, Text
- **XML-Tags** müssen nur mit Buchstaben (Ω , -, ., auch gültig) anfangen
- **Entitäten** in XML sind z.B. `<`, `>`, `"`; für das `<` `>` und `"` `-` Zeichen. (`&`; `)` sind "Fluchtsymbole" in XML. Dadurch können alle Metazeichen in XML auch verwendet werden (`&` wird durch `&` gesetzt)
- `<![CDATA[<svg></svg>]]>` ermöglicht Textabschnitt ohne Fluchtsymbole
- **Kommentare:** `<!-- Kommentar -->`
- **Processing Instruction:** `<?xml-stylesheet href="person.css" type="text/css" ?>` helfen, XML-Dokumente mit temporären Zusatzinformationen zu versehen, die für die Weiterverarbeitung wichtig sind

9.1 DTD

- DTD ist selbst kein XML, gibt Aufbau (oder Syntaxbeschreibung) der XML-Daten mit Regeln an
- Beispieldefinition:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE person [
  <!ELEMENT person(fname,profession*)>
  <!ELEMENT profession(#PCDATA)>
  <!ELEMENT fname(#PCDATA)>
  <!ATTLIST fname id CDATA #REQUIRED>
]>
```

```
<person>
  <fname id="123">Tim</fname>
  <profession>Troll</profession>
  <profession>Toll</profession>
</person>
```

- `<!ELEMENT (...)>` definiert ein Element und in Klammern die erwarteten Kind-Elemente, in fester Reihenfolge. **Anzahl der Nachfolger** muss durch ein Suffix festgelegt werden:
 - ? für ein-oder-keinmal anfügen (0 oder 1),
 - * für beliebig oft oder gar nicht (0 oder ∞),
 - + für ein oder mehr (1 oder ∞).
 Dabei geht **Gruppierung** mit `<!ELEMENT transport (car | plane)>` Als vorgegebenes Kind-Element kann auch `#PCDATA` genommen werden (also beliebig viel normaler Text). Es kann auch `<!ELEMENT br EMPTY>` und in XML `
` statt der Kindelementliste geschrieben werden, wenn das Element immer leer sein soll (oder `<!ELEMENT page ANY>` für komplett egal)
- **Attributtypen** sind z.B. `CDATA` (Text), `NMTOKEN` (wohlgeformter XML-Name), `ENTITY` (Namen einer Entität), `ID` (eindeutiger XML-Name), `IDREF` (Name eines anderen im XML vorkommenden ID Elements), `IDREFS` (Mehrere Namen von anderen XML-Tags im Dokument, getrennt durch Leerzeichen)
- **Voreinstellungen der Attribute:** `<!ATTLIST img source CDATA #REQUIRED` gibt Attributliste für das `img`-Element an. Möglich sind `#REQUIRED` (muss vorhanden sein), `#FIXED` "value" (muss immer einen bestimmten Wert haben), `#IMPLIED` (optional). Alternativ kann eine beliebige Zeichenkette geschrieben werden, die den Standardwert angibt

- **Entitäten:** bestehen aus Ampersand (`&`), einem Entity Namen und Semikolon (`;`)

```
<!ENTITY writer "Donald Duck">
<!ENTITY copyright "Copyright DD">

<author>&writer;&copyright;</author>
```

- **Namespaces:** `<svg xmlns="http://www.w3.org/2000/svg">`, ermöglichen Verwendung verschiedener Formate mit Präfix, um Namenskonflikte zu vermeiden

```
<?xml version="1.0"?>
<html
  xmlns:svg="http://w3.org/2000/svg">
<svg:ellipse rx="1" ry="1"/>
</html>
```

- **XML für Dokumente:** Text Encoding Initiative mit DTD für Textverarbeitung für Theater, Literatur o.ä.
- **DocBook:** XML für Bücher wie Dokumentation von Informatikinhalten
- **OpenEBook** oder **EPUB:** XML-basiertes E-Book-Format für iPad, Kindle etc.
- **MusicXML:** Ein offener Standard für den Austausch von Musik, der die Visualisierung (Noten) und Wiedergabe unterstützt. Leistungsfähiger als MIDI
- **XML Schema:** DTDs können keine Informationen über Datentypen oder maximale Längen für Elemente und Attribute enthalten. Keine Modularisierung, DTDs sind keine XML-Dokumente

10 JSON

JavaScript Object Notation – ist ein Dateiformat in lesbarer Textform, selbstbeschreibend, Verwendung von APIs
Besteht aus `"key": value` Paaren, wobei `"key"` – **String** und `value` Zahl, String, Bool, null, Array `[]` oder Objekt `{}` ist. Kein Kommentar möglich