**XML 🡪 Dokument festMenue**

**Unicode** Das Ziel von Unicode ist es den weltweiten Datenaustausch zu vereinfachen. Einen universalen Zeichensatz, der alle Schriften der Welt umfassen soll, zu erschaffen. Zukünftig auch einen Standard bei den „Emojis“.

**Metasprache für Markupsprachen**

Eine Metasprache ist eine Sprache zum Definieren von Sprachen. Nun, das bedeutet, dass XML die Regeln zur Verfügung stellt mit denen sich dann Sprachen wie zum Beispiel HTML definieren lassen. Will meinen, dass sich mit XML jeder der Lust und Laune hat eine eigene Markup Sprache mit eigenen Tags (oder besser Elementen) erschaffen kann.

**Element** Markierung von Bereichen mit logischen Namen

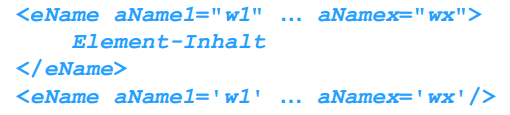
**Attribut** Qualifizierung der Elemente mit Name-Wert-Paaren

**Text** Inhalt

**(eingebettetes) Markup** Textstrom in Unicode/UTF-8 aus Inhaltstext und eingebetteten Tags, die die logische Struktur repräsentieren.

Die Syntax des Kodierungsformats legt fest, wie der Inhaltstext von den Markierungen unterschieden wird (spitze Klammern, Backslash und geschweifte Klammern). Strukturierte Dokumente können auch auf andere (oft proprietäre) Art kodiert werden (siehe Word).

**Markupsprache**: Markiert Textbereiche an Anfang und Ende mit eingebetteten Tags:



spezifisches Vokabular/Format für spezifische Anwendungen, definiert durch DTD oder andere Schemasprache

**Tag**: Tag-Syntax wie bei HTML: Wohlgeformtheit: Tags müssen immer geschlossen werden (auch ohne Inhalt) und können hierarchisch genestelt sein. Die Tags formen Elemente oder Datenbestände.

**Entity-Referenz:** Entities sind definierte Kürzel. Vielleicht kennen Sie aus Textverarbeitungsprogrammen die Möglichkeit, Textbausteine zu definieren, die dann über ein Menü oder Tastaturkürzel sehr schnell in den aktuellen Text einfügbar sind. Genau diese Funktionalität ist eine der Aufgaben von Entities. XML hat genau fünf vordefinierte Entity-Referenzen.

Diese sind:

&lt; 🡪das Kleiner-als-Zeichen, auch als öffnende spitze Klammer bezeichnet (<)

&amp; 🡪das Ampersand (&)

&gt; 🡪das Größer-als-Zeichen, auch als schließende spitze Klammer bezeichnet (>)

&quot; 🡪die geraden doppelten Anführungszeichen (")

&apos; 🡪der Apostroph bzw. das einfache gerade Anführungszeichen (')

**Parser:** Programme zur XML-Syntaxüberprüfung

Aufgaben eines Parsers:

* Syntaxüberprüfung
* Abstraktion von "syntaktischem Zucker"
* Überprüfen der Wohlgeformtheit
* Auswertung von DTD (optional)
  + − Expansion von Referenzen
  + − Einsetzen von Attributwerten
  + − Validierung
* Weitergabe der erforderlichen Information an Anwendung

**Document Type Definition (DTD) ->** Freies Vokabular für Elementnamen, Attributnamen, Referenzen, formal definierbar

Eine Dokumenttypdefinition, ist ein Satz an Regeln, der benutzt wird, um Dokumente eines bestimmten Typs zu deklarieren. Ein Dokumenttyp ist dabei eine Klasse ähnlicher Dokumente, wie beispielsweise Telefonbücher oder Inventurdatensätze.

**Klassifizierung der Begriffe**

Woraus besteht ein XML-Dokument konzeptuell (was sind die Komponenten eines XML-Dokuments)?

* Element, Attribut, Text

Wie ist ein XML-Dokument kodiert?

* (eingebettetes) Markup, Unicode, UTF-8

Welche Einheiten gibt es auf der Kodierungsebene?

* Tag, Entity-Referenz, Kommentare, XML-Deklaration, Processing Instructions

Was ist XML?

* Format (für textuelle Daten, nicht Präsentation), Markupsprache, Metasprache für Markupsprachen

Wie wird ein XML-Dokument verarbeitet?

* Parser, XML-Prozessor

**Übungen:**

* Was stimmt für die XML-Deklaration?
  + A: Muss „Version“, „Encoding“, „Standalone“ enthalten
  + B: Steht am Anfang vom Dokument
  + C: Wird zwingend benötigt
  + D: Enthält Struktur-Elemente
* Was ist das Name-Wertpaar eines Elements?
  + A: Instanz
  + B: Attribut
  + C: Referenz
  + D: Tag
* Was strukturiert ein XML-Dokument?
  + A: Referenzen, Kommentare
  + B: Document Type Definition (DTD), Referenzen
  + C: Elemente, Tags
  + D: CSS-Stylesheet, Tags
* Was ist Wohlgeformtheit?
  + A: Alle Tags hierarchisch genestellt
  + B: Schöne Darstellung im Browser
  + C: Einhaltung der XML-Syntax
  + D: Valide Editierung des Dokuments
* Ein Element kann entweder ein Attribut und/oder weitere Elemente enthalten. Was ist ein Vorteil Attribute zu verwenden?
  + A: Erweiterbarkeit
  + B: Separates Speichern von Metadaten möglich
  + C: Flexibilität
  + D: Attribute können im Gegensatz zu Elementen mehrere Werte enthalten

**Begriffsnetz:**

