Übung 6: Umgang mit DTDs

- Vorgeschichte:
  - Jon Bosak stellte 1992-1994 die Werke von William Shakespeare in SGML-Form zur allgemeinen Verfügung
  - 1996/97 stellte er XML-Versionen bereit
  - Diese Versionen verwendeten eine gemeinsame DTD
  - Dabei sind einige Restriktionen von XML gegenüber SGML offenbar nicht beachtet worden…
- Vorbereitung:
  - cp …/Shakespeare/tempest.xml ueb06.xml # Bereitgestellte
  - cp …/Shakespeare/shaksper.dtd ueb06.dtd # Quellen kopieren

Übung 6: Umgang mit DTDs

- Aufgabe:
  - Vervollständigen Sie die Umstellungsarbeiten von Jon Bosak von SGML zu XML, d.h.: Validieren Sie ueb06.xml
    - Ändern Sie dazu ueb06.dtd und/oder ueb06.xml solange ab, bis die Validierung (mit nsgmls) gelingt.
    - Vorausdenken lohnt sich: Wählen Sie dabei einen Weg, der berücksichtigt, dass Shakespeare noch mehr Stücke als „The Tempest“ geschrieben hat …
  - Vorgehensweise
    - Fehlermeldungen eines nsgmls-Tests - der Reihe nach - auf Ursachen prüfen, diese beseitigen, erneut testen usw.
    - Dabei sollen natürlich die Nutzdaten unverändert bleiben
  - Hinweis:
    - Diese Übung erfordert ein höheres Maß an selbständigem Vorgehen als bisher. Setzen Sie das im Theorieteil erlernte Wissen praktisch ein!
  - Testen Sie nach gelungenem Test mit nsgmls, ob auch mozilla „ueb06.xml“ akzeptiert - das sollte er!

Übung 7: DTD entwickeln (1)

A) Entwickeln einer eigenen DTD

- Datei ueb07.xml enthält eine Instanz (Beispiel) eines Dokumententyps zur Verwaltung von Klausuren. Es fehlt ueb07.dtd

- Vorbereitung:
  - cp …/ueb07.xml ueb07.xml # Bereitgestellte Datei holen
- Aufgabe:
  - Erstellen Sie ueb07.dtd und validieren Sie ueb07.xml
    - Deklarieren Sie Elemente mit sinnvollen Wiederholungseigenschaften
    - Deklarieren Sie alle verwendeten Attribute, ermitteln Sie geeignete Attribut-Typen nach eigenem Ermessen - es gibt nicht nur einen Weg.
    - Experimentieren Sie! Haben Sie den Verdacht, ein Attribut ließe sich auf mehrere verschiedene Wege deklarieren, dann testen Sie die Varianten.
    - Testen Sie den Typ „ID“ für „MatrNr“ des Elements „Student“. Welche Vor- und Nachteile hat diese Variante? Auswege?
  - Testen Sie schließlich, ob auch mozilla „ueb07.xml“ akzeptiert.

Übung 7: DTD entwickeln (2)

B) Modularisierung der DTD

- Das Dokument besteht aus verschiedenen logischen Abschnitten, die unterschiedlichen Zwecken dienen und zu verschiedenen Zeiten und/oder von verschiedenen Personen erstellt werden.
- Einige DTD-Bestandteile bieten sich für eine Mehrfachnutzung an.

- Vorbereitung (dafür eigenes Unterverzeichnis ueb07b anlegen):
  - mkdir ueb07b; cd ueb07b; cp ../ueb07.xml ueb07b.xml; cp ../ueb07.dtd ueb07b.dtd
- Aufgabe:
  - Dokument modularisieren:
    - Lagern Sie aus ueb07b.xml die Elemente Aufgabenliste, Teilnehmerliste, Ergebnisliste und Notenschwellen in je eine externe *entity* aus (aufgabenliste.ent, …, notenschwellen.ent).
  - DTD modularisieren:
    - Lagern Sie die zugehörigen Deklarationen aus in separate *parameter entities*: aufgabenliste-dtd.ent, …, notenschwellen-dtd.ent, zusätzlich: student-dtd.ent
    - Implementieren Sie Attribut „Note“ als *enumeration* mit default „5“, deklarieren Sie die benötigte Notenliste als *parameter entity* in notenauswahl-dtd.ent
  - Validieren Sie Ihr Ergebnis ueb07b.xml mit nsgmls und mozilla.

Übung 8: CSS

- Aufgabe: Entwickeln einer CSS-Datei
  - Wir möchten das Drama „*The Tempest*“ per CSS formatieren.
  - Dies soll schrittweise geschehen, beginnend mit dem einfachen Umbrechen der Inhalte in einzelne Zeilen.
  - Durch Hinzufügen weiterer CSS-Regeln wird die Formatierung schrittweise dem projizierten Beispiel angenähert.
  - Die CSS-Regeln und -Eigenschaften werden nur anfangs bereitgestellt und sind dann eigenständig in den Spezifikationen zu suchen!
- Tips:
  - Gehen Sie methodisch vor: Nehmen Sie sich die Gestaltung ganz bestimmter XML-Elemente vor und setzen Sie diese in die Tat um, bevor Sie weitermachen. Dies reduziert die *trial&error*-Durchgänge.
  - Dies ist keine „Kochbuch“-Übung. Das Zusammensuchen der benötigten Detailinformationen ist Teil der Übung - und entspricht in hohem Maße den Anforderungen im späteren Berufsalltag.

Übung 8: CSS

- Ausgangslage herstellen
  - cp ueb06.xml ueb08.xml # Früheres Übungsergebnis ist Start
  - DTD *entity* anlegen (bereitgestellt): play-css.dtd
  - ueb08.xml anpassen und validieren
  - Anzeige in Browser testen (vgl. Übung 1):
    - Mozilla: Reine *char data*, unbrauchbar
    - IE: Baumstruktur, lesbar aber „ohne“ Formatierung
- PI aufnehmen gemäß Vorlesung (ueb08.css - zunächst *dummy*)
  - Anzeige in Browser testen:
    - Mozilla: Reine *char data*, unverändert
    - IE: Nun analog Mozilla!
- 1) Mini-Stylesheet - eine Zeile in ueb08.css eintragen:
  - * { display: block } /* damit Anzeige in Browser testen */
    - Mozilla und IE: *Char data* - nun aber lesbar, da jedes Element eine neue Zeile (Block) einleitet.

Übung 8: CSS

- *2) Stylesheet*: Ausgewählte Elemente ausblenden
  - fm { display: none }
    - damit Anzeige in Browser testen:
    - Alle Inhalte von <fm>…</fm> (incl. Unterelemente) sind verschwunden!
- *3) Stylesheet*: …
  - (usw, nach und nach die CSS-Datei „imitieren“ / vorstellen)
  - schließlich: Probelauf mit „vollständigem“ *stylesheet*
- Wer schnell vorankommt…
  - … der/die kann weitere Varianten mittels CSS2-Doku ermitteln & ausprobieren:
    - wie Farben funktionieren - Bsp: Grauer Hintergrund für Regieanweisungen!
    - wie man Überschriften zentriert - insb. „Act ...“ und die Szenentitel
    - wie sich konstante Texte hinzufügen lassen: Bsp: Regieanweisungen klammern! - Vergleich Mozilla - IE, hier mit Befund (IE lückenhaft).

Übung 9: CSS2 (Kurzfassung)

Testen zweier sehr nützlicher Eigenschaften, die CSS2 einführte:

- Einfügen konstanter Texte vor / nach Elementen
- Tabellengestaltung

- Ausgangslage herstellen
  - cp ueb03.xml ueb09.xml # Früheres Übungsergebnis ist Start
  - PI anlegen, verweisen Sie auf „ueb09.css“
- Aufgabe
  - Bringen Sie MathAussagen mittels CSS2 als Tabelle zur Anzeige
    - zunächst nur linkeSeite und rechteSeite von MathAussage
  - Wandeln Sie die Attributwerte zu „Typ“ in Ergänzungstext „vor“ rechteSeite um.
    - Hinweis: Beachten Sie die CSS-eigene Codierung von Sonderzeichen.
  - Wandeln Sie die Attributwerte zu „Nummer“ in Ergänzungstext „vor“ MathAussage um, führen Sie dabei Klammern um die Zahlen ein.