

## Übung 10 – Digitale Medien

### Inhalt

XML: XML Dokumente, DTD, XML Schema, Formatierung mittels CSS

### Aufgaben

#### Aufgabe 1: Document Type Definition (DTD)

Erstellen Sie eine DTD zur Darstellung der Raumbellegung an der Uni. Dabei sollen folgende Aspekte berücksichtigt werden:

- Die Räume werden den verschiedenen Gebäuden zugeordnet.
- Ein Gebäude hat eine Adresse, die aus dem Namen der Straße und einer oder mehrerer Hausnummern besteht. Jedes Gebäude enthält mindestens einen Raum.
- Zu jedem Raum gibt es eine eindeutige Raumnummer (ID) und einen Belegungsplan.
- Optional kann ein Raum auch einen Namen erhalten.
- Ein Belegungsplan enthält Wochentage mit einer Abfolge von Belegungszeiten, angegeben durch Beginn und Dauer. Die Dauer kann auch weggelassen werden, dann wird „60“ (Minuten) als Standardwert angenommen.
- Jeder Belegungszeit sind ein Fach und ein Dozent zugeordnet.

#### Aufgabe 2: XML-Dokument

Erstellen Sie beispielhaft eine Übersicht über die Raumbellegung einiger Räume in der Theresienstraße in Form eines XML-Dokuments auf Basis der DTD aus Aufgabe 1. Wann ist das XML-Dokument wohlgeformt und wann ist es gültig?

#### Aufgabe 3: XML Schema

Ergänzen Sie das XML Schema von der Übungs-Webseite um das Element *Wochentag* aus Aufgabe 1.

#### Aufgabe 4 (optional): Anzeige von XML mit Stylesheets

Erstellen Sie eine CSS-Datei, um das XML Dokument aus Aufgabe 2 im Browser formatiert anzuzeigen. Verwenden Sie dazu die erweiterten CSS-Konzepte, wie z.B. `display:block`, sowie `:before` zur Anzeige der Attribute.

Hinweis: Beachten Sie, dass ein Browser benötigt wird, der die benötigten Konzepte auch ausreichend unterstützt (siehe z.B. Angaben bei SelfHTML).

## Anhang – Auswahl von XML-Schema-Komponenten

### Aufbau XML-Schema-Datei

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" {...}>
  {...} //Inhalt
</xs:schema>
```

### Wichtige Schema-Komponenten

```
<element
  maxOccurs = (nonNegativeInteger | unbounded) : 1
  minOccurs = nonNegativeInteger : 1
  name = NCName
  ref = QName
  type = QName
  {...}>
  Content: ({...}, (simpleType | complexType)?, {...})
</element>
```

- maxOccurs: maximale Anzahl
- minOccurs: minimale Anzahl
- NCName: XML-Name ohne „:“
- QName (Qualified Name): Name mit optionalem Namespace-Präfix
- simpleType: Einfacher Typ

```
<complexType
  name = NCName
  {...}>
  Content: ({...} (choice | sequence)?, (attribute | attributeGroup)*, {...})
</complexType>
```

- choice: Auswahl

```
<sequence
  maxOccurs = (nonNegativeInteger | unbounded) : 1
  minOccurs = nonNegativeInteger : 1
  {...}>
  Content: ({...}, (element | choice | sequence | {...})*)
</sequence>
```

```
<attributeGroup
  name = NCName
  ref = QName
  {...}>
  Content: ({...}, (attribute | attributeGroup)*, {...})
</attributeGroup>
```

```
<attribute
  default = string
  name = NCName
  ref = QName
  type = QName
  use = (optional | prohibited | required) : optional
  {...}>
  Content: ({...}, simpleType?)
</attribute>
```

Wert von type für String: "xs:string"

Weitere Informationen: <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>